Data Assimilation:
A Mathematical Introduction

K.J.H. Law, A.M. Stuart and K.C. Zygalakis

May 16, 2015

Abstract

This is an excerpt from the forthcoming book Data Assimilation: A Mathematical Introduction by KJH Law, AM Stuart and KC Zygalakis, to be published by Springer.

Contents

1 Mathematical Background 4

1.1 Probability . . . 4

1.1.1 Random Variables on $\mathbb{R}^\ell$ . . . 4

1.1.2 Gaussian Random Variables . . . 6

1.1.3 Conditional and Marginal Distributions . . . 9

1.1.4 Bayes' Formula . . . 10

1.1.5 Independence . . . 11

1.2 Dynamical Systems . . . 11

1.2.1 Iterated Maps . . . 11

1.2.2 Differential Equations . . . 13

1.2.3 Long-Time Behaviour . . . 15

1.2.4 Controlled Dynamical Systems . . . 16

1.3 Probability Metrics . . . 17

1.3.1 Metric Properties . . . 17

1.3.2 Metrics on Spaces of Probability Measures . . . 18

1.4 Probabilistic View of Dynamical Systems . . . 20

1.4.1 Markov Kernel . . . 21

1.4.2 Ergodicity . . . 22

1.4.3 Bayes' Formula as a Map . . . 23

1.5 Bibliography . . . 24

1.6 Exercises . . . 24

2 Discrete Time: Formulation 26

2.1 Set-Up . . . 26

2.2 Guiding Examples . . . 27

2.3 Smoothing Problem . . . 34

2.3.1 Probabilistic Formulation of Data Assimilation . . . 34

2.3.2 Stochastic Dynamics . . . 34

2.3.3 Reformulation of Stochastic Dynamics . . . 36

2.3.4 Deterministic Dynamics . . . 38

2.4 Filtering Problem . . . 38

2.5 Filtering and Smoothing are Related . . . 39

2.6 Well-Posedness . . . 40

2.7 Assessing The Quality of Data Assimilation Algorithms . . . 44

2.8 Illustrations . . . 46

2.9 Bibliographic Notes . . . 49

2.10 Exercises . . . 52

3 Discrete Time: Smoothing Algorithms **54**
3.1 Linear Gaussian Problems: The Kalman Smoother . . . 55
3.2 Markov Chain-Monte Carlo Methods . . . 58
3.2.1 The MCMC Methodology . . . 58
3.2.2 Metropolis-Hastings Methods . . . 59
3.2.3 Deterministic Dynamics . . . 61
3.2.4 Stochastic Dynamics . . . 62
3.3 Variational Methods . . . 65
3.4 Illustrations . . . 69
3.5 Bibliographic Notes . . . 72
3.6 Exercises . . . 75

4 Discrete Time: Filtering Algorithms **78**
4.1 Linear Gaussian Problems: The Kalman Filter . . . 79
4.2 Approximate Gaussian Filters . . . 81
4.2.1 3DVAR . . . 83
4.2.2 Extended Kalman Filter . . . 83
4.2.3 Ensemble Kalman Filter . . . 83
4.2.4 Square Root Ensemble Kalman Filters . . . 84
4.3 The Particle Filter . . . 86
4.3.1 The Basic Approximation Scheme . . . 86
4.3.2 Sequential Importance Resampling . . . 87
4.3.3 Improved Proposals . . . 92
4.4 Large-Time Behaviour of Filters . . . 94
4.4.1 The Kalman Filter in One Dimension . . . 94
4.4.2 The 3DVAR Filter . . . 97
4.4.3 The Synchronization Filter . . . 99
4.5 Illustrations . . . 100
4.6 Bibliographic Notes . . . 103
4.7 Exercises . . . 111

5 Discrete Time: MATLAB Programs **113**
5.1 Chapter 2 Programs . . . 113
5.1.1 p1.m . . . 113
5.1.2 p2.m . . . 115
5.2 Chapter 3 Programs . . . 117
5.2.1 p3.m . . . 117
5.2.2 p4.m . . . 120
5.2.3 p5.m . . . 122
5.2.4 p6.m . . . 124
5.2.5 p7.m . . . 126
5.3 Chapter 4 Programs . . . 128
5.3.1 p8.m . . . 128
5.3.2 p9.m . . . 131
5.3.3 p10.m . . . 133
5.3.4 p11.m . . . 135
5.3.5 p12.m . . . 137
5.3.6 p13.m . . . 139
5.3.7 p14.m . . . 141
5.3.8 p15.m . . . 143

5.4 ODE Programs 145
5.4.1 p16.m 145
5.4.2 p17.m 145

6 Continuous Time: Formulation 148
6.1 Set-Up 148
6.1.1 Derivation of the Continuous Time Model 148
6.1.2 Stochastic Integration 148
6.2 Guiding Examples 148
6.3 Smoothing Problem 148
6.3.1 Probabilistic Formulation of Data Assimilation 148
6.3.2 Girsanov Formula 148
6.3.3 Conditional Probabilities 148
6.3.4 Stochastic Dynamics 148
6.3.5 Deterministic Dynamics 148
6.4 Filtering Problem 148
6.5 Illustrations 148
6.6 Bibliographic Notes 148
6.7 Exercises 148

7 Continuous Time: Smoothing Algorithms 149
7.1 Linear Gaussian Problems: The Kalman-Bucy Smoother 149
7.2 Markov Chain-Monte Carlo Methods 149
7.2.1 Deterministic Dynamics 149
7.2.2 Stochastic Dynamics 149
7.3 Variational Methods 149
7.4 Illustrations 149
7.5 Bibliographic Notes 149
7.6 Exercises 149

8 Continuous Time: Filtering Algorithms 150
8.1 Linear Gaussian Problems: The Kalman-Bucy Filter 150
8.2 Approximate Gaussian Filters 150
8.2.1 3DVAR 150
8.2.2 Extended Kalman Filter 150
8.2.3 Ensemble Kalman Filter 150
8.3 The Particle Filter 150
8.4 Large-Time Behaviour of Filters 150
8.4.1 The Kalman-Bucy Filter in One Dimension 150
8.4.2 The 3DVAR Filter 150
8.5 Illustrations 150
8.6 Bibliographic Notes 150
8.7 Exercises 150

9 Continuous Time: MATLAB Programs 151
9.1 Chapter 6 Programs 151
9.1.1 p1c.m 151
9.1.2 p2c.m 151
9.1.3 p3c.m 151
9.2 Chapter 7 Programs 151

9.2.1 p4c.m 151
9.2.2 p5c.m 151
9.2.3 p6c.m 151
9.2.4 p7c.m 151
9.3 Chapter 8 Programs 151
9.3.1 p8c.m 151
9.3.2 p10c.m 151
9.3.3 p11c.m 151
9.3.4 p12c.m 151
9.3.5 p13c.m 151
9.3.6 p14c.m 151
9.3.7 p15c.m 151
9.3.8 p16c.m 151
9.3.9 p17c.m 151

Chapter 1

Mathematical Background

The purpose of this chapter is to briefly overview the key mathematical ways of thinking that underpin our presentation of the subject of data assimilation . In particular we touch on the subjects of probability, dynamical systems, probability metrics and dynamical systems for probability measures, in sections 1.1, 1.2, 1.3 and 1.4 respectively. Our treatment is necessarily terse and very selective and the bibliography section 1.5 provides references to the literature. We conclude with exercises in section 1.6.

We highlight here the fact that, throughout this book, all probability measures on $\mathbb{R}^\ell$ will be assumed to possess a density with respect to Lebesgue measure and, furthermore, this density will be assumed to be strictly positive everywhere in $\mathbb{R}^\ell$. This assumption simplifies greatly our subsequent probabilistic calculations.

1.1 Probability

We describe here some basic notation and facts from probability theory, all of which will be fundamental to formulating data assimilation from a probabilistic perspective.

1.1.1. Random Variables on $\mathbb{R}^\ell$

We consider a *random variable* z, defined as a function on an underlying probability space, and taking values in $\mathbb{R}^\ell$. Associated with this random variable is an induced probability measure μ on $\mathbb{R}^\ell$. Furthermore, to be able to compute expectations we need to work with a sufficiently rich collection of subsets of $\mathbb{R}^\ell$, to each of which we can assign the probability that z is contained in it; this collection of subsets is termed a *σ-algebra.* Throughout these notes we work with $\mathcal{B}(\mathbb{R}^\ell)$, the Borel σ-algebra generated by the open sets; we will abbreviate this σ-algebra by $\mathcal{B}$, when the set $\mathbb{R}^\ell$ is clear. The Borel σ-algebra is the natural collection of subsets available on $\mathbb{R}^\ell$ which allows for coherent assignation probabilities, and a theory of integration; an element in $\mathcal{B}$ will be termed a *Borel set.*

We have defined a *probability triple* $\big(\mathbb{R}^\ell, \mathcal{B}, \mu\big)$. For simplicity we assume throughout the book that z has a strictly positive *probability density function* (pdf) ρ with respect to Lebesgue measure.[1] Then, for any Borel set $A \subset \mathbb{R}^\ell$, we define $\mu(A)$, the probability that the random variable z lies in A, by

$$\mu(A) = \int_A \rho(x)\, dx,$$

[1] Of course μ and ρ depend on the particular random variable z, but we suppress this dependence in the notation.

where the pdf $\rho : \mathbb{R}^\ell \to \mathbb{R}^+$ satisfies

$$\int_{\mathbb{R}^\ell} \rho(x)\, dx = 1.$$

A Borel set $A \subset \mathbb{R}^\ell$ is sometimes termed an *event* and the event is said to occur *almost surely* if $\mu(A) = 1$. Since ρ integrates to 1 over $\mathbb{R}^\ell$ and is strictly positive, this implies that the Lebesgue measure of the complement of A, the set A^c, is zero.

We write $z \sim \mu$ as shorthand for the statement that z is *distributed* according to probability measure μ on $\mathbb{R}^\ell$. Note that here $\mu : \mathcal{B}(\mathbb{R}^\ell) \to [0,1]$ denotes a probability measure and $\rho : \mathbb{R}^\ell \to \mathbb{R}^+$ the corresponding density. However, we will sometimes use the letter $\mathbb{P}$ to denote both the measure and its corresponding pdf. This should create no confusion: $\mathbb{P}(\cdot)$ will be a probability measure whenever its argument is a Borel set, and a density whenever its argument is a point in $\mathbb{R}^\ell$. On occasion we will write $\mathbb{P}(z \in A)$ for $\mathbb{P}(A)$.

For any function $f : \mathbb{R}^\ell \to \mathbb{R}^{p\times q}$ we denote by $\mathbb{E} f(z)$ the *expected value* of the random variable $f(z)$ on $\mathbb{R}^{p\times q}$; this expectation is given by

$$\mathbb{E} f(z) = \int_{\mathbb{R}^\ell} f(x)\mu(dx), \quad \mu(dx) = \rho(x)dx.$$

We also sometimes write $\mu(f)$ for $\mathbb{E}f(z)$. The case where the function f is vector valued corresponds to $q = 1$ so that $\mathbb{R}^{p\times q} = \mathbb{R}^{p\times 1} \equiv \mathbb{R}^p$. We will sometimes write $\mathbb{E}^\mu$ if we wish to differentiate between different measures with respect to which the expectation is to be understood.

The *characteristic function* of the random variable z on $\mathbb{R}^\ell$ is $\mathsf{cf} : \mathbb{R}^\ell \to \mathbb{C}$ defined by

$$\mathsf{cf}(h) = \mathbb{E}\exp\bigl(i\langle h, z\rangle\bigr).$$

Example 1.1 *Let $\ell = 1$ and set $\rho(x) = \frac{1}{\pi(1+x^2)}$. Note that $\rho(x) > 0$ for every $x \in \mathbb{R}$. Also, using the change of variables $x = \tan\theta$,*

$$\int_{-\infty}^{\infty} \frac{dx}{\pi(1+x^2)} = 2\int_0^\infty \frac{dx}{\pi(1+x^2)} = \int_{\arctan(0)}^{\arctan(\infty)} \frac{2\sec^2\theta\, d\theta}{\pi(1+\tan^2\theta)} = \frac{2}{\pi}\int_0^{\pi/2} d\theta = 1,$$

and therefore ρ is the pdf of a random variable z on $\mathbb{R}$. We say that such random variable has the Cauchy distribution*.* ♠

Let $G : \mathbb{R}^\ell \to \mathbb{R}^\ell$ and note that if z is a random variable on $\mathbb{R}^\ell$, so too is $G(z)$. If $z \sim \mu$ then $G(z) \sim G \star \mu$, the *pushforward* of μ under G. If μ has associated pdf ρ on $\mathbb{R}^\ell$ then $G \star \mu$ has associated pushforward pdf denoted by $G \star \rho$. This pushforward pdf may be calculated explicitly by means of the change of variable formula under an integral. Indeed if G is invertible then

$$G \star \rho(v) := \rho(G^{-1}(v))|DG^{-1}(v)|.$$

We will occasionally use the *Markov inequality* which states that, for a random variable z on $\mathbb{R}^\ell$, and any $R > 0$,

$$\mathbb{P}(|z| \geq R) \leq R^{-1}\mathbb{E}|z|. \tag{1.1}$$

As a consequence

$$\mathbb{P}(|z| < R) \geq 1 - R^{-1}\mathbb{E}|z|. \tag{1.2}$$

In particular, if $\mathbb{E}|z| < \infty$, then choosing R sufficiently large shows that $\mathbb{P}(|z| < R) > 0$. In our setting this last inequality follows in any case, by assumption on the strict positivity of $\rho(\cdot)$ everywhere on $\mathbb{R}^\ell$.

We say that a sequence of probability measures $\mu^{(n)}$ on $\mathbb{R}^\ell$ is said to *converge weakly* to a limiting probability measures μ on $\mathbb{R}^\ell$ if, for all continuous bounded functions $\varphi : \mathbb{R}^\ell \to \mathbb{R}$,

$$\mathbb{E}^{\mu^{(n)}}\varphi(u) \to \mathbb{E}^{\mu}\varphi(u)$$

as $n \to \infty$.

Finally we note that, although developed here on $\mathbb{R}^\ell$, the theory of probability can be developed on much more general measure spaces and, in particular, on separable Banach space. In the part of the book relating to continuous time we will use probability theory in this setting.

1.1.2. Gaussian Random Variables

We work in finite dimensions, but all the ideas can be generalized to infinite dimensional contexts, such as the separable Hilbert space setting, for example. A *Gaussian* random variable[2] on $\mathbb{R}^\ell$ is characterized by:

- Mean: $m \in \mathbb{R}^\ell$.
- Covariance: $C \in \mathbb{R}^{\ell\times\ell}_{\mathrm{sym}}, \quad C \geq 0$.

We write $z \sim N(m, C)$ and call the Gaussian random variable *centred* if $m = 0$. If $C > 0$ then z has strictly positive pdf on $\mathbb{R}^\ell$, given by

$$\rho(x) = \frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\exp\Big(-\frac{1}{2}\big|C^{-\frac{1}{2}}(x-m)\big|^2\Big) \tag{1.3a}$$

$$= \frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\exp\Big(-\frac{1}{2}|x-m|_C^2\Big). \tag{1.3b}$$

It can be shown that indeed ρ given by (1.3) satisfies

$$\int_{\mathbb{R}^\ell}\rho(x)\,dx = 1. \tag{1.4}$$

Lemma 1.2 *Let $z \sim N(m, C)$, $C > 0$. Then*

1. *$\mathbb{E}z = m$.*
2. *$\mathbb{E}(z-m)(z-m)^T = C$.*

Proof For the first item

$$\begin{aligned}
\mathbb{E}z &= \frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\int_{\mathbb{R}^\ell} x\exp\big(-\frac{1}{2}|x-m|_C^2\big)\,dx\\
&= \frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\int_{\mathbb{R}^\ell} (y+m)\exp\big(-\frac{1}{2}|y|_C^2\big)\,dy\\
&= \frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\int_{\mathbb{R}^\ell} y\exp\big(-\frac{1}{2}|y|_C^2\big)\,dy + \frac{m}{(2\pi)^{\ell/2}(\det C)^{1/2}}\int_{\mathbb{R}^\ell}\exp\big(-\frac{1}{2}|y|_C^2\big)\,dy\\
&= 0 + m\\
&= m,
\end{aligned}$$

[2] Sometimes also called *normal* random variable

where we used in the last line that the function $y \mapsto y\exp\big(-\frac{1}{2}|y|_C^2\big)$ is even and the fact that, by (1.4),

$$\frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\int_{\mathbb{R}^\ell}\exp\big(-\frac{1}{2}|y|_C^2\big)=1.$$

For the second item

$$\begin{aligned}\mathbb{E}(z-m)(z-m)^T&=\frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\int_{\mathbb{R}^\ell}(x-m)(x-m)^T\exp\big(-\frac{1}{2}|x-m|_C^2\big)\,dx\\&=\frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\int_{\mathbb{R}^\ell}yy^T\exp\big(-\frac{1}{2}|C^{-1/2}y|^2\big)\,dy\\&=\frac{1}{(2\pi)^{\ell/2}(\det C)^{1/2}}\int_{\mathbb{R}^\ell}C^{1/2}ww^TC^{1/2}\exp\big(-\frac{1}{2}|w|^2\big)\det(C^{1/2})\,dw\\&=C^{1/2}JC^{1/2}\end{aligned}$$

where

$$J=\frac{1}{(2\pi)^{\ell/2}}\int_{\mathbb{R}^\ell}ww^T\exp\big(-\frac{1}{2}|w|^2\big)\,dw\in\mathbb{R}^\ell\times\mathbb{R}^\ell$$

and so

$$J_{ij}=\frac{1}{(2\pi)^{\ell/2}}\int_{\mathbb{R}^\ell}w_iw_j\exp\big(-\frac{1}{2}\sum_{k=1}^{\ell}w_k^2\big)\prod_{k=1}^{\ell}dw_k.$$

To complete the proof we need to show that J is the identity matrix I on $\mathbb{R}^\ell\times\mathbb{R}^\ell$. Indeed, for $i\neq j$

$$J_{ij}\propto\int_{\mathbb{R}}w_i\exp\big(-\frac{1}{2}w_i^2\big)\,dw_i\int_{\mathbb{R}}w_j\exp\big(-\frac{1}{2}w_j^2\big)\,dw_j=0,$$

by symmetry; and for $i=j$

$$\begin{aligned}J_{jj}&=\frac{1}{(2\pi)^{\frac{1}{2}}}\int_{\mathbb{R}}w_j^2\exp\big(-\frac{1}{2}w_j^2\big)\,dw_j\left(\frac{1}{(2\pi)^{\frac{1}{2}}}\int_{\mathbb{R}}\exp\big(-\frac{1}{2}w_k^2\big)\,dw_k\right)^{\ell-1}\\&=\frac{1}{(2\pi)^{\frac{1}{2}}}\int_{\mathbb{R}}w_j^2\exp\big(-\frac{1}{2}w_j^2\big)\,dw_j\\&=-\frac{1}{(2\pi)^{\frac{1}{2}}}w_j\exp\big(-\frac{1}{2}w_j^2\big)\Big|_{-\infty}^{\infty}+\frac{1}{(2\pi)^{\frac{1}{2}}}\int_{\mathbb{R}}\exp\big(-\frac{1}{2}w_j^2\big)\,dw_j=1,\end{aligned}$$

where we again used (1.4) in the first and last lines. Thus $J=I$, the identity in $\mathbb{R}^\ell$, and $\mathbb{E}(z-m)(z-m)^T=C^{1/2}C^{1/2}=C$. □

The following characterization of Gaussians is often useful.

Lemma 1.3 *The characteristic function of the Gaussian $N(m,C)$ is given by*

$$\mathsf{cf}(h)=\exp\big(i\langle h,m\rangle-\frac{1}{2}\langle Ch,h\rangle\big).$$

Proof This follows from noting that

$$\frac{1}{2}|x-m|_C^2-i\langle h,x\rangle=\frac{1}{2}|x-(m+iCh)|_C^2-i\langle h,m\rangle+\frac{1}{2}\langle Ch,h\rangle.$$

□

Remark 1.4 *Note that the pdf for the Gaussian random variable that we wrote down in equation* (1.3) *is defined only for $C > 0$ since it involves C^{-1}. The characteristic function appearing in the preceding lemma can be used to* define *a Gaussian with mean m and covariance C, including the case where $C \geq 0$ so that the Gaussian covariance C is only positive semi-definite, since it is defined in terms of C and not C^{-1}. For example if we let $z \sim N(m, C)$ with $C = 0$ then z is a* Dirac mass *at m, i.e. $z = m$ almost surely and for any continuous function f*

$$\mathbb{E}f(z) = f(m).$$

This Dirac mass may be viewed as a particular case of a Gaussian random variable. We will write δ_m for $N(m, 0)$. ♠

Lemma 1.5 *The following hold for Gaussian random variables:*

- *If $z = a_1 z_1 + a_2 z_2$ where z_1, z_2 are independent Gaussians with distributions $N(m_1, C_1)$ and $N(m_2, C_2)$ respectively then z is Gaussian with distribution $N(a_1 m_1 + a_2 m_2, a_1^2 C_1 + a_2^2 C_2)$.*
- *If $z \sim N(m, C)$ and $w = Lz + a$ then $w \sim N(Lm + a, LCL^T)$.*

Proof The first result follows from computing the characteristic function of z. By independence this is the product of the characteristic functions of $a_1 z_1$ and of $a_2 z_2$. The characteristic function of $a_i z_i$ has logarithm equal to

$$i\langle h, a_i m_i\rangle - \frac{1}{2}\langle a_i^2 Ch, h\rangle.$$

Adding this for $i = 1, 2$ gives the logarithm of the characteristic function of z, from which its mean and covariance may be read off.

For the second result we note that the characteristic function of $a + Lz$ is the expectation of the exponential of

$$i\langle h, a + Lz\rangle = i\langle h, a\rangle + i\langle L^T h, z\rangle.$$

Using the properties of the characteristic functions of z we deduce that the logarithm of the characteristic function of $a + Lz$ is equal to

$$i\langle h, a\rangle + i\langle L^T h, m\rangle - \frac{1}{2}\langle CL^T h, L^T h\rangle.$$

This may be re-written as

$$i\langle h, a + Lm\rangle - \frac{1}{2}\langle LCL^T h, h\rangle$$

which is the logarithm of the characteristic function of $N(a + Lm, LCL^T)$ as required. □

We finish by stating a lemma whose proof is straightforward, given the foregoing material in this section, and left as an exercise.

Lemma 1.6 *Define*

$$I(v) := \frac{1}{2}\big\langle (v - m), L(v - m)\big\rangle$$

with $L \in \mathbb{R}^{\ell \times \ell}_{\text{sym}}$ satisfying $L > 0$ and $m \in \mathbb{R}^\ell$. Then $\exp\big(-I(v)\big)$ can be normalized to produce the pdf of the Gaussian random variable $N(m, L^{-1})$ on $\mathbb{R}^\ell$. The matrix L is known as the precision matrix *of the Gaussian random variable.*

1.1.3. Conditional and Marginal Distributions

Let $(a, b) \in \mathbb{R}^\ell \times \mathbb{R}^m$ denote a jointly varying random variable.

Definition 1.7 *The* marginal pdf *of a, $\mathbb{P}(a)$, is given in terms of the pdf of (a, b), $\mathbb{P}(a, b)$, by*

$$\mathbb{P}(a) = \int_{\mathbb{R}^m} \mathbb{P}(a, b)\, db.$$

♠

Remark 1.8 *With this definition, for $A \subset \mathcal{B}(\mathbb{R}^\ell)$,*

$$\begin{aligned}\mathbb{P}(a \in A) = \mathbb{P}\Big((a, b) \in A \times \mathbb{R}^m\Big) &= \int_A \int_{\mathbb{R}^m} \mathbb{P}(a, b)\, da\, db \\ &= \int_A \left(\int_{\mathbb{R}^m} \mathbb{P}(a, b)\, db\right) da = \int_A \mathbb{P}(a)\, da.\end{aligned}$$

Thus the marginal pdf $\mathbb{P}(a)$ is indeed the pdf for a in situations where we have no information about the random variable b, other than that it is in $\mathbb{R}^m$. ♠

We now consider the situation which is the extreme opposite of the marginal situation. To be precise, we assume that we know *everything* about the random variable b: we have observed it and know what value it takes. This leads to consideration of the random variable a *given* that we know the value taken by b; we write $a|b$ for a given b. The following definition is then natural:

Definition 1.9 *The* conditional pdf *of $a|b$, $\mathbb{P}(a|b)$, is defined by*

$$\mathbb{P}(a|b) = \frac{\mathbb{P}(a, b)}{\mathbb{P}(b)}. \tag{1.5}$$

♠

Remark 1.10 *Conditioning a jointly varying random variable can be useful when computing probabilities, as the following calculation demonstrates.*

$$\begin{aligned}\mathbb{P}\Big((a, b) \in A \times B\Big) &= \int_A \int_B \mathbb{P}(a, b)\, da\, db \\ &= \int_A \int_B \mathbb{P}(a|b)\mathbb{P}(b)\, da\, db \\ &= \int_B \underbrace{\left(\int_A \mathbb{P}(a|b)\, da\right)}_{=:I_1} \underbrace{\mathbb{P}(b)\, db}_{=:I_2}.\end{aligned}$$

Given b, I_1 computes the probability that a is in A. I_2 then denotes averaging over given outcomes of b in B. ♠

1.1.4. Bayes' Formula

By Definition 1.9 we have

$$\mathbb{P}(a,b) = \mathbb{P}(a|b)\mathbb{P}(b), \tag{1.6a}$$

$$\mathbb{P}(a,b) = \mathbb{P}(b|a)\mathbb{P}(a). \tag{1.6b}$$

Equating and rearranging we obtain **Bayes' formula** which states that

$$\mathbb{P}(a|b) = \frac{1}{\mathbb{P}(b)}\mathbb{P}(b|a)\mathbb{P}(a). \tag{1.7}$$

The importance of this formula is apparent in situations where $\mathbb{P}(a)$ and $\mathbb{P}(b|a)$ are individually easy to write down. Then $\mathbb{P}(a|b)$ may be identified easily too.

Example 1.11 *Let $(a,b) \in \mathbb{R} \times \mathbb{R}$ be a jointly varying random variable specified via*

$$\begin{aligned} a &\sim N(m,\sigma^2), \quad \mathbb{P}(a); \\ b|a &\sim N(f(a),\gamma^2), \quad \mathbb{P}(b|a). \end{aligned}$$

Notice that, by using equation (1.5)*, $\mathbb{P}(a,b)$ is defined via two Gaussian distributions. In fact we have*

$$\mathbb{P}(a,b) = \frac{1}{2\pi\gamma\sigma}\exp\left(-\frac{1}{2\gamma^2}|b-f(a)|^2 - \frac{1}{2\sigma^2}|a-m|^2\right).$$

Unless $f(\cdot)$ is linear this is not the pdf of a Gaussian distribution. Integrating over a we obtain, from the definition of the marginal pdf of b,

$$\mathbb{P}(b) = \frac{1}{2\pi\gamma\sigma}\int_{\mathbb{R}}\exp\left(-\frac{1}{2\gamma^2}|b-f(a)|^2 - \frac{1}{2\sigma^2}|a-m|^2\right)da.$$

Using equation (1.6) *then shows that*

$$\mathbb{P}(a|b) = \frac{1}{\mathbb{P}(b)} \times \frac{1}{2\pi\pi\gamma\sigma}\exp\left(-\frac{1}{2\gamma^2}|b-f(a)|^2 - \frac{1}{2\sigma^2}|a-m|^2\right).$$

Note that $a|b$, like (a,b), is not Gaussian. Thus, for both (a,b) and $a|b$, we have constructed a non-Gaussian pdf in a simple fashion from the knowledge of the two Gaussians and a and $b|a$. ♠

When Bayes' formula (1.7) is used in statistics then typically b is observed data and a is the unknown about which we wish to find information, using the data. In this context we refer to $\mathbb{P}(a)$ as the **prior**, to $\mathbb{P}(b|a)$ as the **likelihood** and to $\mathbb{P}(a|b)$ as the **posterior**. The beauty of Bayes's formula as a tool in applied mathematics is that the likelihood is often easy to determine explicitly, given reasonable assumptions on the observational noise, whilst there is considerable flexibility inherent in modelling prior knowledge via probabilities, to give the prior. Combining the prior and likelihood as in (1.7) gives the posterior, which is the random variable of interest; whilst the probability distributions used to define the likelihood $\mathbb{P}(b|a)$ (via a probability density on the data space) and prior $\mathbb{P}(a)$ (via a probability on the space of unknowns) may be quite simple, the resulting posterior probability distribution can be very complicated. A second key point to note about Bayes' formula in this context is that $\mathbb{P}(b)$, which normalizes the posterior to a pdf, may be hard to determine explicitly, but algorithms exist to find information from the posterior without knowing this normalization constant. We return to this point in subsequent chapters.

1.1.5. Independence

Consider the jointly varying random variable $(a, b) \in \mathbb{R}^\ell \times \mathbb{R}^m$. The random variables a and b are said to be *independent* if

$$\mathbb{P}(a, b) = \mathbb{P}(a)\mathbb{P}(b).$$

In this case, for $f : \mathbb{R}^\ell \to \mathbb{R}^{\ell'}$ and $g : \mathbb{R}^m \to \mathbb{R}^{m'}$,

$$\mathbb{E}f(a)g(b)^T = (\mathbb{E}f(a)) \times \left(\mathbb{E}g(b)^T\right)$$

as

$$\mathbb{E}f(a)g(b)^T = \int_{\mathbb{R}^\ell \times \mathbb{R}^m} f(a)g(b)^T \mathbb{P}(a)\mathbb{P}(b)\, da\, db = \left(\int_{\mathbb{R}^\ell} f(a)\mathbb{P}(a)da\right)\left(\int_{\mathbb{R}^m} g(b)^T\mathbb{P}(b)db\right).$$

An i.i.d. (independent, identically distributed) sequence $\{\xi_j\}_{j\in\mathbb{N}}$ is one for which:[3]

- each ξ_j is distributed according to the same pdf ρ;
- ξ_j is independent of ξ_k for $j \neq k$.

If $\mathbb{J}$ is a subset of $\mathbb{N}$ with finite cardinality then this i.i.d. sequence satisfies

$$\mathbb{P}\left(\{\xi_j\}_{j\in\mathbb{J}}\right) = \prod_{j\in\mathbb{J}} \rho(\xi_j)$$

1.2 Dynamical Systems

We will discuss data assimilation in the context of both discrete-time and continuous-time dynamical systems. In this section we introduce some basic facts about such dynamical systems.

1.2.1. Iterated Maps

Let $\in C(\mathbb{R}^\ell, \mathbb{R}^\ell)$. We will frequently be interested in the iterated map, or discrete-time dynamical system, defined by

$$v_{j+1} = \Psi(v_j), \quad v_0 = u,$$

and in studying properties of the sequence $\{v_j\}_{j\in\mathbb{Z}^+}$. A *fixed point* of the map is a point v_∞ which satisfies $v_\infty = \Psi(v_\infty)$; initializing the map at $u = v_\infty$ will result in a sequence satisfying $v_j = v_\infty$ for all $j \in \mathbb{Z}^+$.

Example 1.12 *Let*

$$\Psi(v) = \lambda v + a.$$

Then

$$v_{j+1} = \lambda v_j + a, \quad v_0 = u.$$

By induction we see that, for $\lambda \neq 1$,

$$v_j = \lambda^j u + a\sum_{i=0}^{j-1} \lambda^i = \lambda^j u + a\frac{1-\lambda^j}{1-\lambda}.$$

[3] This discussion is easily generalized to $j \in \mathbb{Z}^+$.

Thus if $|\lambda| < 1$ then

$$v_j \to \frac{a}{1-\lambda} \quad as \quad j \to \infty.$$

The limiting value $\frac{a}{1-\lambda}$ is a fixed point of the map. ♠

Remark 1.13 *In the preceding example the long-term dynamics of the map, for $|\lambda| < 1$, is described by convergence to a fixed point . Far more complex behaviour is, of course, possible; we will explore such complex behaviour in the next chapter.* ♠

The following result is known as the (discrete time) Gronwall lemma.

Lemma 1.14 *Let $\{v_j\}_{j\in\mathbb{Z}^+}$ be a positive sequence and (λ, a) a pair of reals with $\lambda > 0$. Then if*

$$v_{j+1} \le \lambda v_j + a, \qquad j = 0, 1, \dots$$

it follows that

$$v_j \le \lambda^j v_0 + a\frac{1-\lambda^j}{1-\lambda}, \qquad \lambda \ne 1,$$

and

$$v_j \le v_0 + ja, \qquad \lambda = 1.$$

Proof We prove the case $\lambda \ne 1$ as the case $\lambda = 1$ may be proved similarly. We proceed by induction. The result clearly holds for $j = 0$. Assume that the result is true for $j = J$. Then

$$\begin{aligned} v_{J+1} &\le \lambda v_J + a \\ &\le \lambda\left(\lambda^J v_0 + a\frac{1-\lambda^J}{1-\lambda}\right) + a \\ &= \lambda^{J+1} v_0 + a\frac{\lambda-\lambda^{J+1}}{1-\lambda} + a\frac{1-\lambda}{1-\lambda} \\ &= \lambda^{J+1} v_0 + a\frac{1-\lambda^{J+1}}{1-\lambda}. \end{aligned}$$

This establishes the inductive step and the proof is complete. □

We will also be interested in stochastic dynamical systems of the form

$$v_{j+1} = \Psi(v_j) + \xi_j, \quad v_0 = u,$$

where $\xi = \{\xi_j\}_{j\in\mathbb{N}}$ is an i.i.d. sequence of random variables on $\mathbb{R}^\ell$, and u is a random variable on $\mathbb{R}^\ell$, independent of ξ.

Example 1.15 *This is a simple but important one dimensional (i.e. $\ell = 1$.) example. Let $|\lambda| < 1$ and let*

$$\begin{aligned} v_{j+1} &= \lambda v_j + \xi_j, \qquad \xi_j \sim N(0,\sigma^2) \ \textit{i.i.d.}, \\ v_0 &\sim N(m_0, \sigma_0^2). \end{aligned}$$

By induction

$$v_j = \lambda^j v_0 + \sum_{i=0}^{j-1} \lambda^{j-i-1} \xi_i.$$

Thus v_j is Gaussian, as a linear transformation of Gaussians – see Lemma 1.5. Furthermore, using independence of the initial condition from the sequence ξ, we obtain

$$\begin{aligned} m_j &:= \mathbb{E}v_j = \lambda^j m_0 \\ \sigma_j^2 &:= \mathbb{E}(v_j - m_j)^2 = \lambda^{2j}\mathbb{E}(v_0 - m_0)^2 + \sum_{i=0}^{j-1} \lambda^{2j-2i-2}\sigma^2 \\ &= \lambda^{2j}\sigma_0^2 + \sigma^2 \sum_{i=0}^{j-1} \lambda^{2i} = \lambda^{2j}\sigma_0^2 + \sigma^2 \frac{1-\lambda^{2j}}{1-\lambda^2}. \end{aligned}$$

Since $|\lambda| < 1$ we deduce that $m_j \to 0$ and $\sigma_j^2 \to \sigma^2(1-\lambda^2)^{-1}$. Thus the sequence of Gaussians generated by this stochastic dynamical system has a limit, which is a centred Gaussian with variance larger than the variance of ξ_1, unless $\lambda = 0$. ♠

1.2.2. Differential Equations

Let $f \in C^1(\mathbb{R}^\ell, \mathbb{R}^\ell)$ and consider the ordinary differential equation (ODE)

$$\frac{dv}{dt} = f(v), \qquad v(0) = u.$$

Assume a solution exists for all $u \in \mathbb{R}^\ell$, $t \in \mathbb{R}^+$; for any given u this solution is then an element of the space $C^1(\mathbb{R}^+; \mathbb{R}^\ell)$. In this situation, the ODE generates a continuous-time dynamical system. We are interested in properties of the function v. An *equilibrium point* $v_\infty \in \mathbb{R}^\ell$ is a point for which $f(v_\infty) = 0$. Initializing the equation at $u = v_\infty$ results in a solution $v(t) = v_\infty$ for all $t \geq 0$.

Example 1.16 *Let $f(v) = -\alpha v + \beta$. Then*

$$e^{\alpha t}\left(\frac{dv}{dt} + \alpha v\right) = \beta e^{\alpha t}$$

and so

$$\frac{d}{dt}\left(e^{\alpha t} v\right) = \frac{d}{dt}\left(\frac{\beta}{\alpha} e^{\alpha t}\right).$$

Thus

$$e^{\alpha t} v(t) - u = \frac{\beta}{\alpha}(e^{\alpha t} - 1),$$

so that

$$v(t) = e^{-\alpha t} u + \frac{\beta}{\alpha}(1 - e^{-\alpha t}).$$

If $\alpha > 0$ then

$$v(t) \to \frac{\beta}{\alpha} \qquad \textit{as } t \to \infty.$$

Note that $v_\infty := \frac{\beta}{\alpha}$ is a the unique equilibrium point of the equation. ♠

Remark 1.17 *In the preceding example the long-term dynamics of the ODE, for $\alpha > 0$, is described by convergence to an equilibrium point . As in discrete time, far more complex behaviour is, of course, possible; we will explore this possibility in the next chapter.* ♠

If the differential equation has a solution for every $u \in \mathbb{R}^\ell$ and every $t \in \mathbb{R}^+$ then there is a one-parameter semigroup of operators $\Psi(\cdot;t)$, parametrized by time $t \geq 0$, with the properties that

$$\begin{aligned} v(t) &= \Psi(u;t),\ t \in (0,\infty), && (1.10a)\\ \Psi(u;t+s) &= \Psi\big(\Psi(u;s);t\big),\ t,s \in \mathbb{R}^+, u \in \mathbb{R}^\ell, && (1.10b)\\ \Psi(u;0) &= u \in \mathbb{R}^\ell. && (1.10c) \end{aligned}$$

We call $\Psi(\cdot;\cdot)$ the solution operator for the ODE. In this scenario we can consider the iterated map defined by $\Psi(\cdot) = \Psi(\cdot;h)$, for some fixed $h > 0$, thereby linking the discrete time iterated maps with continuous time ODEs.

Example 1.18 *(Example 1.16 continued) Let*

$$\Psi(u;t) = e^{-\alpha t}u + \frac{\beta}{\alpha}\Big(1 - e^{-\alpha t}\Big)$$

which is the solution operator for the equation in that $v(t) = \Psi(u;t)$*. Clearly* $\Psi(u;0) = u$*. Also*

$$\begin{aligned} \Psi(u;t+s) &= e^{-\alpha t}e^{-\alpha s}u + \frac{\beta}{\alpha}\Big(1 - e^{-\alpha t}e^{-\alpha s}\Big)\\ &= e^{-\alpha t}\left(e^{-\alpha s}u + \frac{\beta}{\alpha}(1 - e^{-\alpha s})\right) + \frac{\beta}{\alpha}\Big(1 - e^{-\alpha t}\Big)\\ &= \Psi\left(\Psi(u;s);t\right). \end{aligned}$$

♠

The following result is known as the (continuous time) Gronwall lemma.

Lemma 1.19 *Let* $z \in C^1(\mathbb{R}^+, \mathbb{R})$ *satisfy*

$$\frac{dz}{dt} \leq az + b, \quad z(0) = z_0,$$

for some $a, b \in \mathbb{R}$*. Then*

$$z(t) \leq e^{at}z_0 + \frac{b}{a}\big(e^{at} - 1\big).$$

Proof Multiplying both sides of the given identity by e^{-at} we obtain

$$e^{-at}\left(\frac{dz}{dt} - az\right) \leq be^{-at}$$

which implies that

$$\frac{d}{dt}\left(e^{-at}z\right) \leq be^{-at}.$$

Therefore,

$$e^{-at}z(t) - z(0) \leq \frac{b}{a}\big(1 - e^{-at}\big)$$

so that

$$z(t) \leq e^{at}z_0 + \frac{b}{a}\big(e^{at} - 1\big).$$

□

1.2.3. Long-Time Behaviour

We consider the long-time behaviour of discrete-time dynamical systems . The ideas are easily generalized to continuous-time dynamical systems – ODEs – and indeed our example will demonstrate such a generalization. To facilitate our definitions we now extend Ψ to act on Borel subsets of $\mathbb{R}^\ell$. Note that currently $\Psi : \mathbb{R}^\ell \to \mathbb{R}^\ell$; we extend to $\Psi : \mathcal{B}(\mathbb{R}^\ell) \to \mathcal{B}(\mathbb{R}^\ell)$ via

$$\Psi(A) = \bigcup_{u \in A} \Psi(u), \quad A \in \mathcal{B}(\mathbb{R}^\ell).$$

For both $\Psi : \mathbb{R}^\ell \to \mathbb{R}^\ell$ and $\Psi : \mathcal{B}(\mathbb{R}^\ell) \to \mathcal{B}(\mathbb{R}^\ell)$ we denote by

$$\Psi^{(j)} = \Psi \circ \cdots \circ \Psi$$

the $j-$fold composition of Ψ with itself. In the following, let $B(0, R)$ denote the ball of radius R in $\mathbb{R}^\ell$, in the Euclidean norm, centred at the origin.

Definition 1.20 *A discrete time dynamical system has a bounded* absorbing set $\mathcal{B}_{\text{abs}} \subset \mathbb{R}^\ell$ *if, for every $R > 0$, there exists $J = J(R)$ such that*

$$\Psi^{(J)}\left(B(0, R)\right) \subset \mathcal{B}_{\text{abs}}, \quad \forall j \geq J.$$

♠

Remark 1.21 *The definition of absorbing set is readily generalized to continuous time dynamical systems; this is left as an exercise for the reader.* ♠

Example 1.22 *Consider an ODE for which there exist $\alpha, \beta > 0$ such that*

$$\langle f(v), v \rangle \leq \alpha - \beta |v|^2, \quad \forall v \in \mathbb{R}^\ell.$$

Then

$$\frac{1}{2}\frac{d}{dt}|v|^2 = \left\langle v, \frac{dv}{dt} \right\rangle = \langle v, f(v) \rangle \leq \alpha - \beta |v|^2.$$

Applying the Gronwall Lemma 1.19 gives

$$|v(t)|^2 \leq e^{-2\beta t}|v(0)|^2 + \frac{\alpha}{\beta}\big(1 - e^{-2\beta t}\big).$$

Hence, if $|v(0)|^2 \leq R$ then

$$|v(t)|^2 \leq 2\frac{\alpha}{\beta} \quad \forall t \geq T : e^{-2\beta t}R^2 \leq \frac{\alpha}{\beta}.$$

Therefore the set $\mathcal{B}_{\text{abs}} = B\left(0, \sqrt{\frac{2\alpha}{R}}\right)$ is absorbing for the ODE (with the generalization of the above definition of absorbing set to continuous time, as in Remark 1.21).

If $v_j = v(jh)$ so that $\Psi(\cdot) = \Psi(\cdot; h)$ and $v_{j+1} = \Psi(v_j)$ then

$$|v_j|^2 \leq 2\frac{\alpha}{\beta} \quad \forall J \geq \frac{T}{h},$$

where T is as for the ODE case. Hence $\mathcal{B}_{\text{abs}} = B\left(0, \sqrt{\frac{2\alpha}{R}}\right)$ is also an absorbing set for the iterated map associated with the ODE. ♠

Definition 1.23 *When the discrete time dynamical system has a bounded absorbing set $\mathcal{B}_{\text{abs}}$ we define the global attractor $\mathcal{A}$ to be*

$$\mathcal{A} = \bigcap_{k\ge 0} \overline{\bigcup_{j\ge k} \Psi^{(j)}(\mathcal{B}_{\text{abs}})}.$$

♠

This object captures all the long-time dynamics of the dynamical system. As for the absorbing set itself this definition is readily generalized to continuous time.

1.2.4. Controlled Dynamical Systems

It is frequently of interest to add a *controller* $w = \{w_j\}_{j=0}^{\infty}$ to the discrete time dynamical system to obtain

$$v_{j+1} = \Psi(v_j) + w_j.$$

The aim of the controller is to "steer" the dynamical system to achieve some objective. Interesting examples include:

- given point $v^* \in \mathbb{R}^\ell$ and time $J \in \mathbb{Z}^+$, choose w so that $v_J = v^*$;
- given open set B and time $J \in \mathbb{Z}^+$, choose w so that $v_j \in B$ for all $j \ge J$;
- given $y = \{y_j\}_{j\in\mathbb{N}}$, where $y_j \in \mathbb{R}^m$, and given a function $h : \mathbb{R}^\ell \to \mathbb{R}^m$, choose w to keep $|y_j - h(v_j)|$ small in some sense.

The third option is most relevant in the context of data assimilation , and so we focus on it. In this context we will consider controllers of the form $w_j = K\big(y_j - h(v_j)\big)$ so that

$$v_{j+1} = \Psi(v_j) + K\big(y_j - h(v_j)\big). \tag{1.11}$$

A key question is then how to choose K to ensure the desired property. We present a simple example which illustrates this.

Example 1.24 *Let $\ell = m = 1$, $\Psi(v) = \lambda v$ and $h(v) = v$. We assume that the data $\{y_j\}_{j\in\mathbb{N}}$ is given by $y_{j+1} = v^\dagger_{j+1}$ where $v^\dagger_{j+1} = \lambda v^\dagger_j$. Thus the data is itself generated by the uncontrolled dynamical system. We wish to use the controller to ensure that the solution of the controlled system is close to the data $\{y_j\}_{j\in\mathbb{N}}$ generated by the uncontrolled dynamical system, and hence to the solution of the uncontrolled dynamical system itself.*

Consider the controlled dynamical system

$$\begin{aligned} v_{j+1} &= \Psi(v_j) + K\left(y_j - h(v_j)\right) \\ &= \lambda v_j + \underbrace{K(y_j - v_j)}_{w_j}, \qquad j \ge 1. \end{aligned}$$

and assume that $v_0 \ne v^\dagger_0$. We are interested in whether v_j approaches $v^\dagger_j$ as $j \to \infty$.

To this end suppose that K is chosen so that $|\lambda - K| < 1$. Then note that

$$v^\dagger_{j+1} = \lambda v^\dagger_j + K\underbrace{\left(y_j - v^\dagger_j\right)}_{=0}.$$

Hence $e_j = v_j - v_j^\dagger$ satisfies

$$e_{j+1} = (\lambda - K)e_j$$

and

$$|e_{j+1}| = |\lambda - K||e_j|.$$

Since we have chosen K so that $|\lambda - K| < 1$ then we have $|e_j| \to 0$ as $j \to \infty$. Thus the controlled dynamical system approaches the solution of the uncontrolled dynamical system as $j \to \infty$. This is prototypical of certain data assimilation algorithms that we will study in Chapter 4. ♠

It is also of interest to consider continuous time controllers $\{w(t)\}_{t\geq 0}$ for differential equations

$$\frac{dv}{dt} = f(v) + w.$$

Again, the goal is to choose w to achieve some objective analogous to those described in discrete time.

1.3 Probability Metrics

Since we will frame data assimilation in terms of probability, natural measures of robustness of the problem will require the idea of distance between probability measures. Here we introduce basic metric properties, and then some specific distances on probability measures, and their properties.

1.3.1. Metric Properties

Definition 1.25 *A metric on a set X is a function $d : X \times X \to \mathbb{R}^+$ (distance) satisfying the following properties:*

- *coincidence: $d(x, y) = 0$ iff $x = y$;*
- *symmetry: $d(x, y) = d(y, x)$;*
- *triangle: $d(x, z) \leq d(x, y) + d(y, z)$.*

♠

Example 1.26 *Let $X = \mathbb{R}^\ell$ viewed as a normed vector space with norm $\|\cdot\|$; for example we might take $\|\cdot\| = |\cdot|$, the Euclidean norm. Then the function $d : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}^+$ given by $d(x, y) := \|x - y\|$ defines a metric. Indeed*

- *$\|x - y\| = 0$ iff $x = y$.*
- *$\|x - y\| = \|y - x\|$.*
- *$\|x - z\| = \|x - y + y - z\| \leq \|x - y\| + \|y - z\|$.*

from properties of norms. ♠

1.3.2. Metrics on Spaces of Probability Measures

Let $\mathcal{M}$ denote the space of probability measures on $\mathbb{R}^\ell$ with strictly positive Lebesgue density on $\mathbb{R}^\ell$. Throughout this section we let μ and μ' be two probability measures on $\mathcal{M}$, and let ρ and ρ' denote the corresponding densities; recall that we assume that these densities are positive everywhere, in order to simplify the presentation. We define two useful metrics on probability measures.

Definition 1.27 *The* total variation distance *on $\mathcal{M}$ is defined by*

$$\begin{aligned} d_{\mathrm{TV}}(\mu,\mu') &= \frac{1}{2}\int_{\mathbb{R}^\ell} |\rho(u)-\rho'(u)|\,du \\ &= \frac{1}{2}\mathbb{E}^\mu\left|1-\frac{\rho'(u)}{\rho(u)}\right|. \end{aligned}$$

♠

Thus the total variation distance is half of the L^1 norm of the difference of the two pdfs. Note that clearly $d_{\mathrm{TV}}(\mu,\mu') \geq 0$. Also

$$\begin{aligned} d_{\mathrm{TV}}(\mu,\mu') &\leq \frac{1}{2}\int_{\mathbb{R}^\ell} |\rho(u)|\,du + \frac{1}{2}\int_{\mathbb{R}^\ell} |\rho'(u)|\,du \\ &= \frac{1}{2}\int_{\mathbb{R}^\ell} \rho(u)\,du + \frac{1}{2}\int_{\mathbb{R}^\ell} \rho'(u)\,du \\ &= 1. \end{aligned}$$

Note also that d_{TV} may be characterized as

$$d_{\mathrm{TV}}(\mu,\mu') = \frac{1}{2}\mathrm{sup}_{|f|_\infty\leq 1}|\mathbb{E}^\mu(f)-\mathbb{E}^{\mu'}(f)| = \frac{1}{2}\mathrm{sup}_{|f|_\infty\leq 1}|\mu(f)-\mu'(f)| \tag{1.12}$$

where we have used the convention that $\mu(f)=\mathbb{E}^\mu(f)=\int_{\mathbb{R}^\ell} f(v)\mu(dv)$ and $|f|_\infty = \sup_u |f(u)|$.

Definition 1.28 *The* Hellinger distance *on $\mathcal{M}$ is defined by*

$$\begin{aligned} d_{\mathrm{Hell}}(\mu,\mu') &= \left(\frac{1}{2}\int_{\mathbb{R}^\ell}\left(\sqrt{\rho(u)}-\sqrt{\rho'(u)}\right)^2 du\right)^{1/2} \\ &= \left(\frac{1}{2}\mathbb{E}^\mu\left(1-\sqrt{\frac{\rho'(u)}{\rho(u)}}\right)^2\right)^{1/2}. \end{aligned}$$

♠

Thus the Hellinger distance is a multiple of the L^2 distance between the square-roots of the two pdfs. Again clearly $d_{\mathrm{Hell}}(\mu,\mu') \geq 0$. Also

$$d_{\mathrm{Hell}}(\mu,\mu')^2 \leq \frac{1}{2}\int_{\mathbb{R}^\ell}\left(\rho(u)+\rho'(u)\right)du = 1.$$

We also note that the Hellinger and TV distances can be written in a symmetric way and satisfy the triangle inequality – they are indeed valid distance metrics on the space of probability measures.

Lemma 1.29 *The total variation and Hellinger distances satisfy*

$$0 \le \frac{1}{\sqrt{2}} d_{\mathrm{TV}}(\mu, \mu') \le d_{\mathrm{Hell}}(\mu, \mu') \le d_{\mathrm{TV}}(\mu, \mu')^{1/2} \le 1$$

Proof The upper and lower bounds of, respectively, 0 and 1 are proved above. We show first that $\frac{1}{\sqrt{2}} d_{\mathrm{TV}}(\mu, \mu') \le d_{\mathrm{Hell}}(\mu, \mu')$. Indeed, by the Cauchy-Schwarz inequality,

$$\begin{aligned}
d_{\mathrm{TV}}(\mu, \mu') &= \frac{1}{2} \int_{\mathbb{R}^\ell} \left|1 - \sqrt{\frac{\rho'(u)}{\rho(u)}}\right| \left|1 + \sqrt{\frac{\rho'(u)}{\rho(u)}}\right| \rho(u)\, du \\
&\le \left(\frac{1}{2} \int_{\mathbb{R}^\ell} \left|1 - \sqrt{\frac{\rho'(u)}{\rho(u)}}\right|^2 \rho(u)\, du \right)^{1/2} \left(\frac{1}{2} \int_{\mathbb{R}^\ell} \left|1 + \sqrt{\frac{\rho'(u)}{\rho(u)}}\right|^2 \rho(u)\, du \right)^{1/2} \\
&\le d_{\mathrm{Hell}}(\mu, \mu') \left(\int_{\mathbb{R}^\ell} \left|1 + \frac{\rho'(u)}{\rho(u)}\right| \rho(u)\, du \right)^{1/2} \\
&= \sqrt{2} d_{\mathrm{Hell}}(\mu, \mu').
\end{aligned}$$

Finally, for the inequality $d_{\mathrm{Hell}}(\mu, \mu') \le d_{\mathrm{TV}}(\mu, \mu')^{1/2}$ note that

$$|\sqrt{a} - \sqrt{b}| \le \sqrt{a} + \sqrt{b} \quad \forall a, b > 0.$$

Therefore,

$$\begin{aligned}
d_{\mathrm{Hell}}(\mu, \mu')^2 &= \frac{1}{2} \int_{\mathbb{R}^\ell} \left|1 - \sqrt{\frac{\rho'(u)}{\rho(u)}}\right| \left|1 - \sqrt{\frac{\rho'(u)}{\rho(u)}}\right| \rho(u)\, du \\
&\le \frac{1}{2} \int_{\mathbb{R}^\ell} \left|1 - \sqrt{\frac{\rho'(u)}{\rho(u)}}\right| \left|1 + \sqrt{\frac{\rho'(u)}{\rho(u)}}\right| \rho(u)\, du \\
&= \frac{1}{2} \int_{\mathbb{R}^\ell} \left|1 - \frac{\rho'(u)}{\rho(u)}\right| \rho(u)\, du \\
&= d_{\mathrm{TV}}(\mu, \mu').
\end{aligned}$$

□

Why do we bother to introduce the Hellinger distance, rather than working with the more familiar total variation? The answer stems from the following two lemmas.

Lemma 1.30 *Let* $f : \mathbb{R}^\ell \to \mathbb{R}^p$ *be such that*

$$(\mathbb{E}^\mu |f(u)|^2 + \mathbb{E}^{\mu'} |f(u)|^2) < \infty.$$

Then

$$|\mathbb{E}^\mu f(u) - \mathbb{E}^{\mu'} f(u)| \le 2(\mathbb{E}^\mu |f(u)|^2 + \mathbb{E}^{\mu'} |f(u)|^2)^{\frac{1}{2}} d_{\mathrm{Hell}}(\mu, \mu'). \tag{1.13}$$

As a consequence

$$|\mathbb{E}^\mu f(u) - \mathbb{E}^{\mu'} f(u)| \le 2(\mathbb{E}^\mu |f(u)|^2 + \mathbb{E}^{\mu'} |f(u)|^2)^{\frac{1}{2}} d_{\mathrm{tv}}(\mu, \mu')^{\frac{1}{2}}. \tag{1.14}$$

Proof In the following all integrals are over $\mathbb{R}^\ell$. Now

$$\begin{aligned}
|\mathbb{E}^{\mu} f(u) - \mathbb{E}^{\mu'} f(u)| &\leq \int |f(u)||\rho(u) - \rho'(u)|du \\
&= \int \sqrt{2}|f(u)||\sqrt{\rho(u)} + \sqrt{\rho'(u)}| \cdot \frac{1}{\sqrt{2}}|\sqrt{\rho(u)} - \sqrt{\rho'(u)}|du \\
&\leq \left(\int 2|f(u)|^2|\sqrt{\rho(u)} + \sqrt{\rho'(u)}|^2 du\right)^{\frac{1}{2}} \left(\frac{1}{2}\int |\sqrt{\rho(u)} - \sqrt{\rho'(u)}|^2 du\right)^{\frac{1}{2}} \\
&\leq \left(\int 4|f(u)|^2(\rho(u) + \rho'(u))du\right)^{\frac{1}{2}} \left(\frac{1}{2}\int \left(1 - \frac{\sqrt{\rho'(u)}}{\sqrt{\rho(u)}}\right)^2 \rho(u)du\right)^{\frac{1}{2}} \\
&= 2(\mathbb{E}^{\mu}|f(u)|^2 + \mathbb{E}^{\mu'}|f(u)|^2)^{\frac{1}{2}} d_{\text{Hell}}(\mu, \mu').
\end{aligned}$$

Thus (1.13) follows. The bound (1.14) follows from Lemma 1.29. □

Remark 1.31 *The preceding lemma shows that, if two measures μ and μ' are $\mathcal{O}(\epsilon)$ close in the Hellinger metric, and if the function $f(u)$ is square integrable with respect to u distributed according to μ and μ', then expectations of $f(u)$ with respect to μ and μ' are also $\mathcal{O}(\epsilon)$ close. It also shows that, under the same assumptions on f, if two measures μ and μ' are $\mathcal{O}(\epsilon)$ close in the total variation metric, then expectations of $f(u)$ with respect to μ and μ' are only $\mathcal{O}(\epsilon^{\frac{1}{2}})$ close. This second result is sharp and to get $\mathcal{O}(\epsilon)$ closeness of expectations using $\mathcal{O}(\epsilon)$ closeness in the TV metric requires a stronger assumption on f, as we now show.* ♠

Lemma 1.32 *Assume that $|f|$ is finite almost surely with respect to both μ and μ' and denote the almost sure upper bound on $|f|$ by $f_{\max}$. Then*

$$|\mathbb{E}^{\mu} f(u) - \mathbb{E}^{\mu'} f(u)| \leq 2 f_{\max} d_{\text{TV}}(\mu, \mu').$$

Proof Under the given assumption on f,

$$\begin{aligned}
|\mathbb{E}^{\mu} f(u) - \mathbb{E}^{\mu'} f(u)| &\leq \int |f(u)||\rho(u) - \rho'(u)|du \\
&\leq 2 f_{\max}\left(\frac{1}{2}\int |\rho(u) - \rho'(u)|du\right) \\
&\leq 2 f_{\max}\left(\frac{1}{2}\int \left|1 - \frac{\rho'(u)}{\rho(u)}\right| \rho(u)du\right) \\
&= 2 f_{\max} d_{\text{TV}}(\mu, \mu').
\end{aligned}$$

□

The implication of the preceding two lemmas and remark is that it is natural to work with the Hellinger metric, rather than the total variation metric, whenever considering the effect of perturbations of the measure on expectations of functions which are square integrable, but not bounded.

1.4 Probabilistic View of Dynamical Systems

Here we look at the natural connection between dynamical systems, and the underlying dynamical system that they generate on probability measures. The key idea here is that the

Markovian propagation of probability measures is linear, even when the underlying dynamical system is nonlinear. This advantage of linearity is partially offset by the fact that the underlying dynamics on probability distributions is infinite dimensional, but it is nonetheless a powerful perspective on dynamical systems. Example 1.15 provides a nice introductory example demonstrating the probability distributions carried by a stochastic dynamical system; in that case the probability distributions are Gaussian and we explicitly characterize their evolution through the mean and covariance. The idea of mapping probability measures under the dynamical system can be generalized, but is typically more complicated because the probability distributions are typically not Gaussian and not characterized by a finite number of parameters.

1.4.1. Markov Kernel

Definition 1.33 *$p : \mathbb{R}^\ell \times \mathcal{B}(\mathbb{R}^\ell) \to \mathbb{R}^+$ is a* Markov kernel *if:*

- *for each $x \in \mathbb{R}^\ell$, $p(x, \cdot)$ is a probability measure on $\big(\mathbb{R}^\ell, \mathcal{B}(\mathbb{R}^\ell)\big)$;*
- *$x \mapsto p(x, A)$ is $\mathcal{B}(\mathbb{R}^\ell)$-measurable for all $A \in \mathcal{B}(\mathbb{R}^\ell)$.*

♠

The first condition is the key one for the material in this book: the Markov kernel at fixed x describes the probability distribution of a new point $y \sim p(x, \cdot)$. By iterating on this we may generate a sequence of points which constitute a sample from the distribution of the Markov chain, as described below, defined by the Markov kernel. The second measurability condition ensures an appropriate mathematical setting for the problem, but an in-depth understanding of this condition is not essential for the reader of this book. In the same way that we use $\mathbb{P}$ to denote both the probability measure and its pdf, we sometimes use $p(x, \cdot) : \mathbb{R}^\ell \to \mathbb{R}^+$, for each fixed $x \in \mathbb{R}^\ell$, to denote the corresponding pdf of the Markov kernel from the preceding definition.

Consider the stochastic dynamical system

$$v_{j+1} = \Psi(v_j) + \xi_j,$$

where $\xi = \{\xi_j\}_{j \in \mathbb{Z}^+}$ is an i.i.d. sequence distributed according to probability measure on $\mathbb{R}^\ell$ with density $\rho(\cdot)$. We assume that the initial condition v_0 is possibly random, but independent of ξ. Under these assumptions on the probabilistic structure, we say that $\{v_j\}_{j \in \mathbb{Z}^+}$ is a *Markov chain.* For this Markov chain we have

$$\mathbb{P}(v_{j+1}|v_j) = \rho\big(v_{j+1} - \Psi(v_j)\big);$$

thus

$$\mathbb{P}(v_{j+1} \in A|v_j) = \int_A \rho\big(v_{j+1} - \Psi(v_j)\big)\, dv.$$

In fact we can define a Markov Kernel

$$p(u, A) = \int_A \rho\big(v - \Psi(u)\big)\, dv,$$

with the associated pdf

$$p(u, v) = \rho\left(v - \Psi(u)\right).$$

If $v_j \sim \mu_j$ with pdf ρ_j then

$$\begin{aligned}\mu_{j+1} &= \mathbb{P}(v_{j+1} \in A) \\ &= \int_{\mathbb{R}^\ell} \mathbb{P}(v_{j+1} \in A|v_j)\mathbb{P}(v_j)\, dv_j \\ &= \int_{\mathbb{R}^\ell} p(u, A)\rho_j(u)\, du.\end{aligned}$$

And then

$$\begin{aligned}\rho_{j+1}(v) &= \int_{\mathbb{R}^\ell} p(u, v)\rho_j(u)\, du \\ &= \int_{\mathbb{R}^\ell} \rho\big(v - \Psi(u)\big)\rho_j(u)\, du.\end{aligned}$$

Furthermore we have a linear dynamical system for the evolution of the pdf

$$\rho_{j+1} = P\rho_j, \tag{1.15}$$

where P is the integral operator

$$(P\pi)(v) = \int_{\mathbb{R}^\ell} \rho\big(v - \Psi(u)\big)\pi(u)\, du.$$

Example 1.34 *Let $\Psi : \mathbb{R}^\ell \to \mathbb{R}^\ell$. Assume that $\xi_1 \sim N(0, \sigma^2 I)$. Then*

$$\rho_{j+1}(v) = \int_{\mathbb{R}^\ell} \frac{1}{(2\pi)^{\ell/2}\sigma^\ell} \exp\left(-\frac{1}{2\sigma^2}|v - \Psi(u)|^2\right) \rho_j(u)\, du.$$

As $\sigma \to \infty$ we obtain the deterministic model

$$\rho_{j+1}(v) = \int_{\mathbb{R}^\ell} \delta\big(v - \Psi(u)\big)\rho_j(u)\, du.$$

♠

For each integer $n \in \mathbb{N}$, we use the notation $p^n(u, \cdot)$ to denote the Markov kernel arising from n steps of the Markov chain; thus $p^1(u, \cdot) = p(u, \cdot)$. Furthermore, $p^n(u, A) = \mathbb{P}(u^{(n)} \in A|u^{(0)} = u)$.

1.4.2. Ergodicity

In many situations we will appeal to *ergodic theorems* to extract information from sequences $\{v_j\}_{j\in\mathbb{Z}^+}$ generated by a (possibly stochastic) dynamical systems. Assume that this dynamical systems is invariant with respect to probability measure μ_∞. Then, roughly speaking, an ergodic dynamical systems is one for which, for a suitable class of test functions $\varphi : \mathbb{R}^\ell \to \mathbb{R}$, and v_0 almost surely with respect to the *invariant measure* μ_∞, the Markov chain from the previous subsection satisfies

$$\frac{1}{J}\sum_{j=1}^{J} \varphi(v_j) \to \int_{\mathbb{R}^\ell} \varphi(v)\mu_\infty(dv) = \mathbb{E}^{\mu_\infty}\varphi(v). \tag{1.16}$$

We say that *the time average equals the space average.* The preceding identity encodes the idea that the histogram formed by a single trajectory $\{v_j\}$ of the Markov chain looks more and more like the pdf of the underlying invariant measure. Since the convergence is almost sure with respect to the initial condition, this implies that the statistics of where the trajectory spends time is, asymptotically, independent of the initial condition; this is a very powerful property.

If the Markov chain has a unique *invariant density* ρ_∞, which is a fixed point of the linear dynamical system (1.15), then it will satisfy

$$\rho_\infty = P\rho_\infty, \tag{1.17}$$

or equivalently

$$\rho_\infty(v) = \int_{\mathbb{R}^\ell} p(u,v)\rho_\infty(u)du. \tag{1.18}$$

In the ergodic setting, this equation will have a form of uniqueness within the class of pdfs and, furthermore, it is often possible to prove, in some norm, the convergence

$$\rho_j \to \rho_\infty \quad \text{as } j \to \infty.$$

Example 1.35 *Example 1.15 generates an ergodic Markov chain $\{v_j\}_{j\in\mathbb{Z}^+}$ carrying the sequence of pdfs ρ_j. Furthermore, each ρ_j is the density of a Gaussian $N(m_j,\sigma_j^2)$. If $|\lambda|<1$ then $m_j\to 0$ and $\sigma_j^2\to\sigma_\infty^2$ where*

$$\sigma_\infty^2 = \frac{\sigma^2}{1-\lambda^2}.$$

Thus ρ_∞ is the density of a Gaussian $N(0,\sigma_\infty^2)$. We then have,

$$\frac{1}{J}\sum_{j=1}^{J}\varphi(v_j) = \frac{1}{J}\sum_{j=1}^{J}\varphi\left(\lambda^j v_0 + \sum_{i=1}^{J-1}\lambda^{j-i-1}\xi_i\right) \to \int_{\mathbb{R}} \rho_\infty(v)\varphi(v)\,dv.$$

♠

1.4.3. Bayes' Formula as a Map

Recall that Bayes' formula states that

$$\mathbb{P}(a|b) = \frac{1}{\mathbb{P}(b)}\mathbb{P}(b|a)\mathbb{P}(a).$$

This may be viewed as a map from $\mathbb{P}(a)$ (what we know about a a priori, the prior) to $\mathbb{P}(a|b)$ (what we know about a once we have observed the variable b, the posterior.) Since

$$\mathbb{P}(b) = \int_{\mathbb{R}^\ell} \mathbb{P}(b|a)\mathbb{P}(a)\,da$$

we see that

$$\mathbb{P}(a|b) = \frac{\mathbb{P}(b|a)\mathbb{P}(a)}{\int_{\mathbb{R}^\ell}\mathbb{P}(b|a)\mathbb{P}(a)\,da} =: L\mathbb{P}(a).$$

L is a *nonlinear* map which takes pdf $\mathbb{P}(a)$ into $\mathbb{P}(a|b)$. We use the letter L to highlight the fact that the map is defined, in the context of Bayesian statistics, by using the likelihood to map prior to posterior.

1.5 Bibliography

- For background material on probability, as covered in section 1.1, the reader is directed to the elementary textbook [20], and to the more advanced texts [50, 120] for further material (for example the definition of measurable.) The book [92], together with the references therein, provides an excellent introduction to Markov chains. The book [87] is a comprehensive study of ergodicity for Markov chains; the central use of Lyapunov functions will make it particular accessible for readers with a background in dynamical systems. Note also that Theorem 3.3 contains a basic ergodic result for Markov chains.
- Section 1.2 concerns dynamical systems and stochastic dynamical systems. The deterministic setting is over-viewed in numerous textbooks, such as [51, 119], with more advanced material, related to infinite dimensional problems, covered in [109]. The ergodicity of stochastic dynamical systems is over-viewed in [7] and targeted treatments based on the small noise scenario include [43, 12]. The book [107] contains elementary chapters on dynamical systems, and the book chapter [59] contains related material in the context of stochastic dynamical systems. For the subject of control theory the reader is directed to [121], which has a particularly good exposition of the linear theory, and [104] for the nonlinear setting.
- Probability metrics are the subject of section 1.3 and the survey paper [47] provides a very readable introduction to this subject, together with references to the wider literature.
- Viewing (stochastic) dynamical systems as generating a dynamical system on the probability measure which they carry is an enormously powerful way of thinking. The reader is directed to the books [117] and [8] for overviews of this subject, and further references.

1.6 Exercises

1. Consider the ODE
$$\frac{dv}{dt} = v - v^3, \quad v(0) = v_0.$$
By finding the exact solution, determine the one-parameter semigroup $\Psi(\cdot; t)$ with properties (1.10).

2. Consider a jointly varying random variable $(a, b) \in \mathbb{R}^2$ defined as follows: $a \sim N(0, \sigma^2)$ and $b|a \sim N(a, \gamma^2)$. Find a formula for the probability density function of (a, b), using (1.5b), and demonstrate that the random variable is a Gaussian with mean and covariance which you should specify. Using (1.7), find a formula for the probability density function of $a|b$; again demonstrate that the random variable is a Gaussian with mean and covariance which you should specify.

3. Consider two Gaussian densities on $\mathbb{R}$: $N(m_1, \sigma_1^2)$ and $N(m_2, \sigma_2^2)$. Show that the Hellinger distance between them is given by
$$d_{\text{Hell}}(\mu, \mu')^2 = 1 - \sqrt{\exp\Big(-\frac{(m_1 - m_2)^2}{2(\sigma_1^2 + \sigma_2^2)}\Big)\frac{2\sigma_1\sigma_2}{(\sigma_1^2 + \sigma_2^2)}}.$$

4. Consider two Gaussian measures on $\mathbb{R}$: $N(m_1, \sigma_1^2)$ and $N(m_2, \sigma_2^2)$. Show that the total variation distance between the measures tends to zero if $m_2 \to m_1$ and $\sigma_2^2 \to \sigma_1^2$.

5. The Kullback-Leibler divergence between two measures μ' and μ, with pdfs ρ' and ρ respectively, is

$$D_{\text{KL}}(\mu'||\mu) = \int \log\Big(\frac{\rho(x)'}{\rho(x)}\Big)\rho'(x)dx.$$

Does D_{KL} define a metric on probability measures? Justify your answer. Consider two Gaussian densities on $\mathbb{R}$: $N(m_1, \sigma_1^2)$ and $N(m_2, \sigma_2^2)$. Show that the Kullback-Leibler divergence between them is given by

$$D_{\text{KL}}(\mu_1||\mu_2) = \ln\Big(\frac{\sigma_2}{\sigma_1}\Big) + \frac{1}{2}\Big(\frac{\sigma_1^2}{\sigma_2^2} - 1\Big) + \frac{(m_2 - m_1)^2}{2\sigma_2^2}.$$

6. Assume that two measures μ and μ' have positive Lebesgue densities ρ and ρ' respectively. Prove the bounds:

$$d_{\text{Hell}}(\mu, \mu')^2 \leq \frac{1}{2} D_{\text{KL}}(\mu||\mu') \ , \qquad d_{\text{TV}}(\mu, \mu')^2 \leq D_{\text{KL}}(\mu||\mu'),$$

where the Kullback-Leibler divergence D_{KL} is defined in the preceding exercise.

7. Consider the stochastic dynamical system of Example 1.15. Find explicit formulae for the maps $m_j \mapsto m_{j+1}$ and $\sigma_j^2 \mapsto \sigma_{j+1}^2$.

8. Directly compute the mean and covariance of $w = a + Lz$ if z is Gaussian $N(m, C)$, without using the characteristic function. Verify that you obtain the same result as in Lemma 1.5.

9. Prove Lemma 1.6.

10. Generalize Definitions 1.20 and 1.23 to continuous time, as suggested in Remark 1.21.

Chapter 2

Discrete Time: Formulation

In this chapter we introduce the mathematical framework for discrete-time data assimilation. Section 2.1 describes the mathematical models we use for the underlying signal, which we wish to recover, and for the data, which we use for the recovery. In section 2.2 we introduce a number of examples used throughout the text to illustrate the theory. Sections 2.3 and 2.4 respectively describe two key problems related to the conditioning of the signal v on the data y, namely **smoothing** and **filtering**; in section 2.5 we describe how these two key problems are related. Section 2.6 proves that the smoothing problem is well-posed and, using the connection to filtering described in 2.5, that the filtering problem is well-posed; here well-posedness refers to continuity of the desired conditioned probability distribution with respect to the observed data. Section 2.7 discusses approaches to evaluating the quality of data assimilation algorithms. In section 2.8 we describe various illustrations of the foregoing theory and conclude the chapter with section 2.9 devoted to a bibliographical overview and section 2.10 containing exercises.

2.1 Set-Up

We assume throughout the book that $\Psi \in C(\mathbb{R}^n, \mathbb{R}^n)$ and consider the Markov chain $v = \{v_j\}_{j\in\mathbb{Z}^+}$ defined by the random map

$$v_{j+1} = \Psi(v_j) + \xi_j,\ j \in \mathbb{Z}^+, \tag{2.1a}$$

$$v_0 \sim N(m_0, C_0), \tag{2.1b}$$

where $\xi = \{\xi_j\}_{j\in\mathbb{Z}^+}$ is an i.i.d. sequence, with $\xi_0 \sim N(0, \Sigma)$ and $\Sigma > 0$. Because (v_0, ξ) is a random variable, so too is the solution sequence $\{v_j\}_{j\in\mathbb{Z}^+}$: the **signal**, which determines the state of the system at each discrete time instance. For simplicity we assume that v_0 and ξ are independent. The probability distribution of the random variable v quantifies the uncertainty in predictions arising from this **stochastic dynamics** model.

In many applications, models such as (2.1) are supplemented by observations of the system as it evolves; this information then changes the probability distribution on the signal, typically reducing the uncertainty. To describe such situations we assume that we are given **data**, or **observations**, $y = \{y_j\}_{j\in\mathbb{N}}$ defined as follows. At each discrete time instance we observe a (possibly nonlinear) function of the signal, with additive noise :

$$y_{j+1} = h(v_{j+1}) + \eta_{j+1},\ j \in \mathbb{Z}^+, \tag{2.2}$$

where $h \in C(\mathbb{R}^n, \mathbb{R}^m)$ and $\eta = \{\eta_j\}_{j \in \mathbb{N}}$ is an i.i.d. sequence, independent of (v_0, ξ), with $\eta_1 \sim N(0, \Gamma)$ and $\Gamma > 0$. The function h is known as the **observation operator**. The objective of **data assimilation** is to determine information about the signal v, given data y. Mathematically we wish to solve the problem of conditioning the random variable v on the observed data y, or problems closely related to this. Note that we have assumed that both the model noise ξ and the observational noise η are Gaussian; this assumption is made for convenience only, and could be easily generalized.

We will also be interested in the case where the dynamics is deterministic and (2.1) becomes

$$v_{j+1} = \Psi(v_j), \ j \in \mathbb{Z}^+, \tag{2.3a}$$

$$v_0 \sim N(m_0, C_0). \tag{2.3b}$$

In this case, which we refer to as **deterministic dynamics**, we are interested in the random variable v_0, given the observed data y; note that v_0 determines all subsequent values of the signal v.

Finally we mention that in many applications the function Ψ is the solution operator for an ordinary differential equation (ODE) of the form [1]

$$\frac{dv}{dt} = f(v), \ t \in (0, \infty), \tag{2.4a}$$

$$v(0) = v_0. \tag{2.4b}$$

Then, assuming the solution exists for all $t \geq 0$, there is a one-parameter semi-group of operators $\Psi(\cdot; t)$, parametrized by time $t \geq 0$, with properties defined in (1.10). In this situation we assume that $\Psi(u) = \Psi(u; \tau)$, i.e. the solution operator over τ time units, where τ is the time between observations; thus we implicitly make the simplifying assumption that the observations are made at equally spaced time-points, and note that the state $v_j = v(jh)$ evolves according to (2.3a). We use the notation $\Psi^{(j)}(\cdot)$ to denote the $j-$fold composition of Ψ with itself. Thus, in the case of continuous time dynamics, $\Psi(\cdot\,; j\tau) = \Psi^{(j)}(\cdot)$.

2.2 Guiding Examples

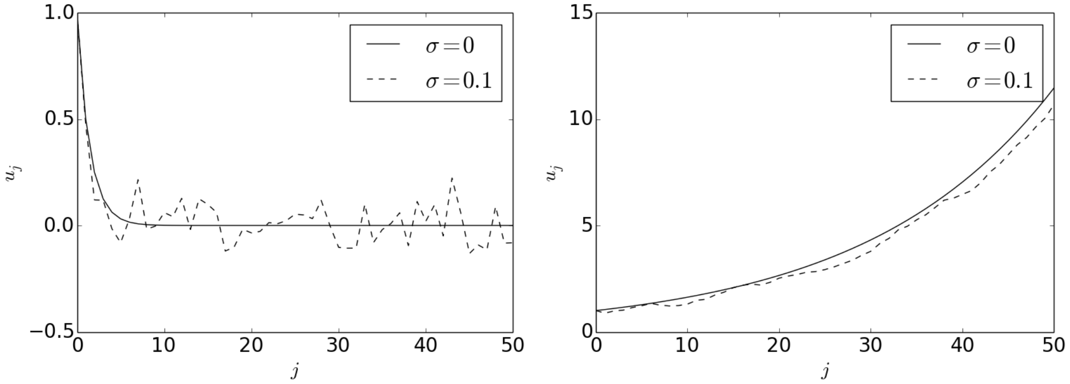
Throughout these notes we will use the following examples to illustrate the theory and algorithms presented.

Example 2.1 *We consider the case of one dimensional linear dynamics where*

$$\Psi(v) = \lambda v \tag{2.5}$$

for some scalar $\lambda \in \mathbb{R}$. Figure 2.1 compares the behaviour of the stochastic dynamics (2.1) and deterministic dynamics (2.3) for the two values $\lambda = 0.5$ and $\lambda = 1.05$. We set $\Sigma = \sigma^2$ and in both cases 50 iterations of the map are shown. We observe that the presence of noise does not significantly alter the dynamics of the system for the case when $|\lambda| > 1$, since for both the stochastic and deterministic models $|v_j| \to \infty$ as $j \to \infty$. The effects of stochasticity are more pronounced when $|\lambda| < 1$, since in this case the deterministic map satisfies $v_j \to 0$ whilst, for the stochastic model, v_j fluctuates randomly around 0.

[1] Here the use of $v = \{v(t)\}_{t \geq 0}$ for the solution of this equation should be distinguished from our use of $v = \{v_j\}_{j=0}^{\infty}$ for the solution of (2.1).



(a) $\lambda = 0.5$

(b) $\lambda = 1.05$

Figure 2.1: Behaviour of (2.1) for Ψ given by (2.5) for different values of λ and $\Sigma = \sigma^2$.

Using (2.1a), together with the linearity of Ψ *and the Gaussianity of the noise* ξ_j*, we obtain*

$$\mathbb{E}(v_{j+1}) = \lambda \mathbb{E}(v_j), \quad \mathbb{E}(v_{j+1}^2) = \lambda^2 \mathbb{E}(v_j^2) + \sigma^2.$$

If $|\lambda| > 1$ *then the the second moment explodes as* $j \to \infty$*, as does the modulus of the first moment if* $\mathbb{E}(v_0) \neq 0$*. On the other hand, if* $|\lambda| < 1$*, we see (Example 1.15) that* $\mathbb{E}(v_j) \to 0$ *and* $\mathbb{E}(v_j^2) \to \sigma_\infty^2$ *where*

$$\sigma_\infty^2 = \frac{\sigma^2}{1 - \lambda^2}. \tag{2.6}$$

Indeed, since v_0 *is Gaussian, the model (2.1a) with linear* Ψ *and Gaussian noise* ξ_j *gives rise to a random variable* v_j *which is also Gaussian. Thus, from the convergence of the mean and the second moment of* v_j*, we conclude that* v_j *converges weakly to the random variable* $N(0, \sigma_\infty^2)$*. This is an example of ergodicity as expressed in* (1.16)*; the invariant measure* μ_∞ *is the Gaussian* $N(0, \sigma_\infty^2)$ *and the density* ρ_∞ *is the Lebesgue density of this Gaussian.* ♠

Example 2.2 *Now consider the case of two dimensional linear dynamics. In this case*

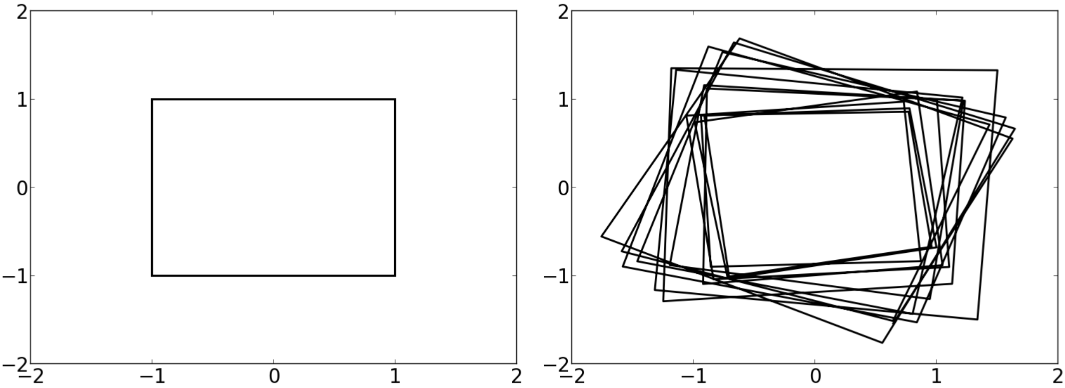
$$\Psi(v) = Av, \tag{2.7}$$

with A *a* 2×2 *dimensional matrix of one of the following three forms* A_ℓ*:*

$$A_1 = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} \lambda & \alpha \\ 0 & \lambda \end{pmatrix}, \quad A_3 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

For $\ell = 1, 2$ *the behaviour of (2.1) for* $\Psi(u) = A_\ell u$ *can be understood from the analysis underling the previous Example 2.1 and the behaviour is similar, in each coordinate, depending on whether the* λ *value on the diagonal is smaller than, or larger than,* 1*. However, the picture is more interesting when we consider the third choice* $\Psi(u) = A_3 u$ *as, in this case, the matrix* A_3 *has purely imaginary eigenvalues and corresponds to a rotation by* $\pi/2$ *on the plane; this is illustrated in Figure 2.2a. Addition of noise into the dynamics gives a qualitatively different picture: now the step* j *to* $j+1$ *corresponds to a rotation by* $\pi/2$*, composed with a random shift of origin; this is illustrated in Figure 2.2b.*

♠



(a) Deterministic dynamics

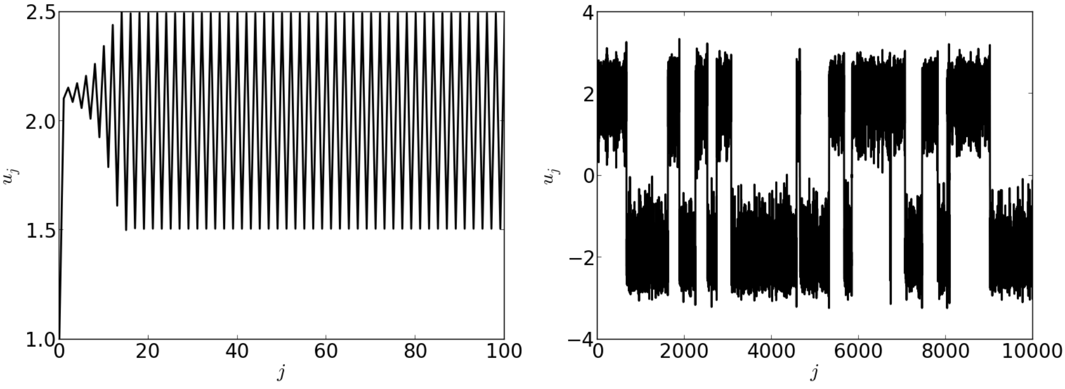
(b) Stochastic dynamics, $\sigma = 0.1$

Figure 2.2: Behaviour of (2.1) for Ψ given by (2.7), and $\Sigma = \sigma^2$.

Example 2.3 *We now consider our first nonlinear example, namely the one-dimensional dynamics for which*

$$\Psi(v) = \alpha \sin v. \tag{2.8}$$

Figure 2.3 illustrates the behaviour of (2.1) for this choice of Ψ, and with $\alpha = 2.5$, both for deterministic and stochastic dynamics . In the case of deterministic dynamics, Figure 2.3a, we see that eventually iterates of the discrete map converge to a period 2 solution. Although only one period 2 solution is seen in this single trajectory, we can deduce that there will be another period 2 solution, related to this one by the symmetry $u \mapsto -u$. This second solution is manifest when we consider stochastic dynamics . Figure 2.3b demonstrates that the inclusion of noise significantly changes the behaviour of the system. The signal now exhibits bistable behaviour and, within each mode of the behavioural dynamics, vestiges of the period 2 dynamics may be seen: the upper mode of the dynamics is related to the period 2 solution shown in Figure 2.3a and the lower mode to the period 2 solution found from applying the symmetry $u \mapsto -u$ to obtain a second period 2 solution from that shown in Figure 2.3a.



(a) Deterministic dynamics

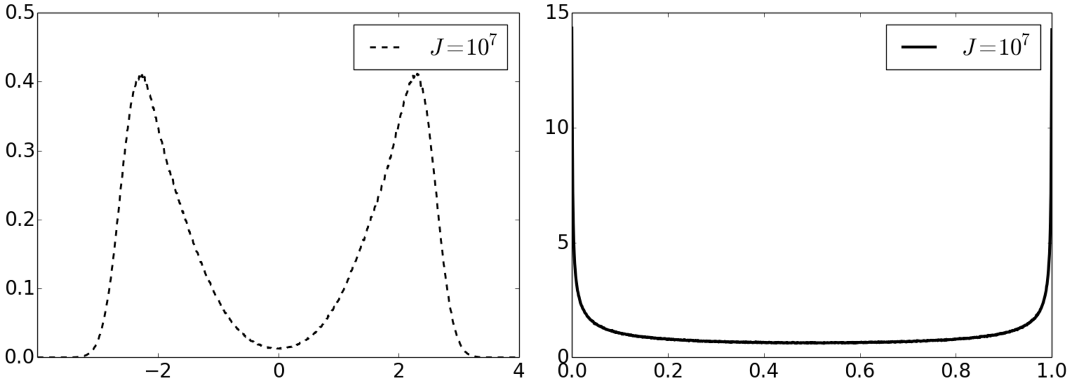
(b) Stochastic dynamics, $\sigma = 0.25$

Figure 2.3: Behaviour of (2.1) for Ψ given by (2.8) for $\alpha = 2.5$ and $\Sigma = \sigma^2$, see also `p1.m` in section 5.1.1.

A good way of visualizing ergodicity is via the empirical measure *or* histogram, *generated by a trajectory of the dynamical system. Equation* (1.16) *formalizes the idea that the histogram, in the large J limit, converges to the probability density function of a random variable, independently of the starting point v_0. Thinking in terms of pdfs of the signal, or functions of the signal, and neglecting time-ordering information, is a very useful viewpoint throughout these notes.*

Histograms visualize complex dynamical behaviour such as that seen in Figure 2.3b by ignoring time-correlation in the signal and simply keeping track of where *the solution goes as time elapses, but not the* order *in which places are visited. This is illustrated in Figure 2.4a, where we plot the histogram corresponding to the dynamics shown in Figure 2.3b, but calculated using a simulation of length $J = 10^7$. We observe that the system quickly forgets its initial condition and spends an almost equal proportion of time around the positive and negative period* 2 *solutions of the underlying deterministic map. The Figure 2.4a would change very little if the system were started from a different initial condition, reflecting ergodicity of the underlying map.*

♠



(a) Example 2.3, as in Figure 2.3b

(b) Example 2.4, as in Figure 2.5b

Figure 2.4: Probability density functions for $v_j, j = 0, \cdots, J$, for $J = 10^7$

Example 2.4 *We now consider a second one-dimensional and nonlinear map, for which*

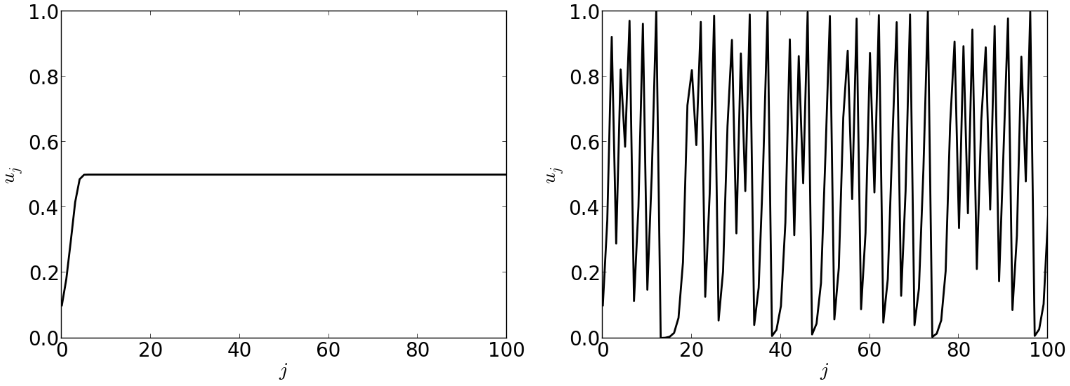
$$\Psi(v) = rv(1 - v). \tag{2.9}$$

We consider initial data $v_0 \in [0, 1]$ noting that, for $r \in [0, 4]$, the signal will then satisfy $v_j \in [0, 1]$ for all j, in the case of the deterministic dynamics (2.3)*. We confine our discussion here to the deterministic case which can itself exhibit quite rich behaviour. In particular, the behaviour of (2.3, 2.9) can be seen in Figure 2.5 for the values of $r = 2$ and $r = 4$. These values of r have the desirable property that it is possible to determine the signal analytically. For $r = 2$ one obtains*

$$v_j = \frac{1}{2} - \frac{1}{2}(1 - 2v_0)^{2^j}, \tag{2.10}$$

which implies that, for any value of $v_0 \neq 0, 1$, $v_j \to 1/2$ as we can also see in Figure 2.5a. For $v_0 = 0$ the solution remains at the unstable fixed point 0*, whilst for $v_0 = 1$ the solution maps onto* 0 *in one step, and then remains there. In the case $r = 4$ the solution is given by*

$$v_j = 4\sin^2(2^j\pi\theta), \quad \textit{with} \quad v_0 = 4\sin^2(\pi\theta) \tag{2.11}$$



(a) Deterministic dynamics, $r = 2$ (b) Deterministic dynamics, $r = 4$

Figure 2.5: Behaviour of (2.1) for Ψ given by (2.9).

This solution can also be expressed in the form

$$v_j = \sin^2(2\pi z_j). \tag{2.12}$$

where

$$z_{j+1} = \begin{cases} 2z_j, & 0 \le z_j < \frac{1}{2}, \\ 2z_j - 1, & \frac{1}{2} \le z_j < 1, \end{cases}$$

and using this formula it is possible to show that this map produces chaotic dynamics for almost all initial conditions. This is illustrated in Figure 2.5b, where we plot the first 100 *iterations of the map. In addition, in Figure 2.4b, we plot the pdf using a long trajectory of* v_j *of length* $J = 10^7$*, demonstrating the ergodicity of the map. In fact there is an analytic formula for the steady state value of the pdf (the invariant density) found as* $J \to \infty$*; it is given by*

$$\rho(x) = \pi^{-1}x^{-1/2}(1-x)^{-1/2}. \tag{2.13}$$

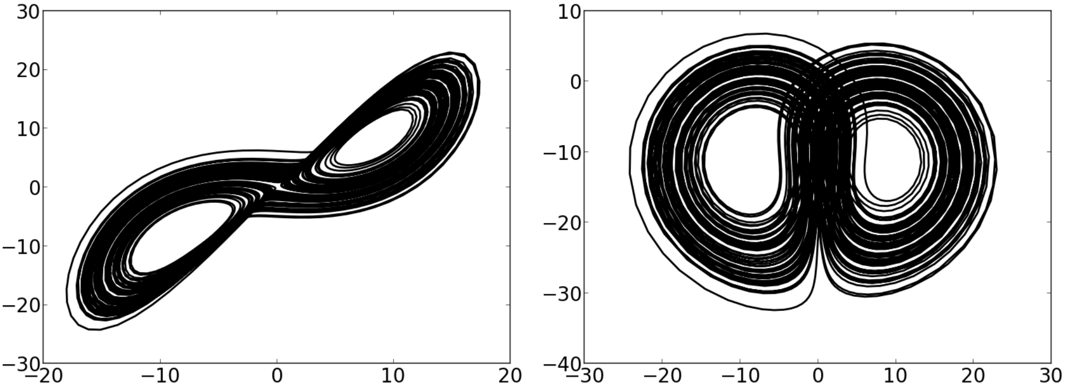
♠

Example 2.5 *Turning now to maps* Ψ *derived from differential equations, the simplest case is to consider linear autonomous dynamical systems of the form*

$$\frac{dv}{dt} = Lv, \tag{2.14a}$$
$$v(0) = v_0. \tag{2.14b}$$

Then $\Psi(u) = Au$ *with* $A = \exp(L\tau)$. ♠

Example 2.6 *The Lorenz '63 model is perhaps the simplest continuous-time system to exhibit sensitivity to initial conditions and chaos. It is a system of three coupled non-linear ordinary*



(a) u_1 vs u_2

(b) u_2 vs u_3

Figure 2.6: Projection of the Lorenz'63 attractor onto two different pairs of coordinates.

differential equations whose solution $v \in \mathbb{R}^3$, where $v = (v_1, v_2, v_3)$, satisfies[2]

$$\begin{aligned} \frac{dv_1}{dt} &= a(v_2 - v_1), && (2.15a)\\ \frac{dv_2}{dt} &= -av_1 - v_2 - v_1v_3, && (2.15b)\\ \frac{dv_3}{dt} &= v_1v_2 - bv_3 - b(r + a). && (2.15c) \end{aligned}$$

Note that we have employed a coordinate system where the origin in the original version of the equations proposed by Lorenz is shifted. In the coordinate system that we employ here we have equation (2.4) with vector field f satisfying

$$\langle f(v), v\rangle \le \alpha - \beta |v|^2 \tag{2.16}$$

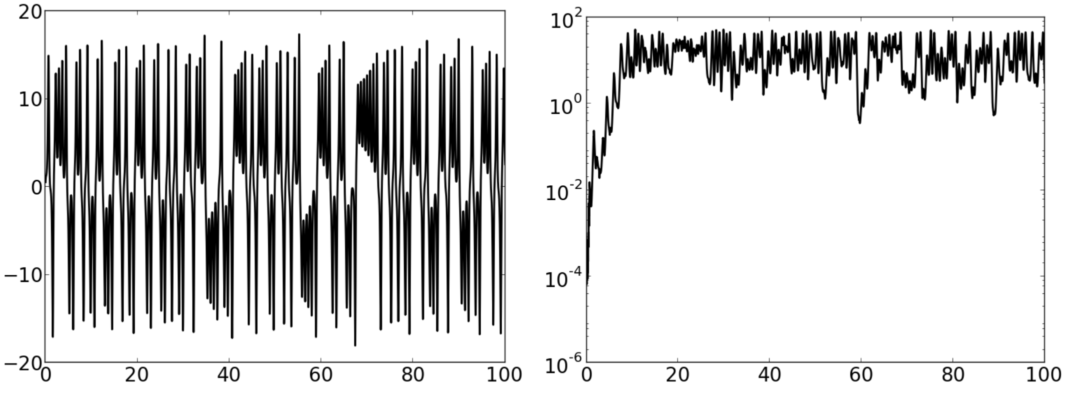
for some $\alpha, \beta > 0$. As demonstrated in Example 1.22, this implies the existence of an absorbing set:

$$\limsup_{t\to\infty} |v(t)|^2 < R \tag{2.17}$$

for any $R > \alpha/\beta$. Mapping the ball $B(0, R)$ forward under the dynamics gives the global attractor (see Definition 1.23) for the dynamics. In Figure 2.6 we visualize this attractor, projected onto two different pairs of coordinates at the classical parameter values $(a, b, r) = (10, \frac{8}{3}, 28)$.

Throughout these notes we will use the classical parameter values $(a, b, r) = (10, \frac{8}{3}, 28)$ in all of our numerical experiments; at these values the system is chaotic and exhibits sensitive dependence with respect to the initial condition. A trajectory of v_1 versus time can be found in Figure 2.7a and in Figure 2.7b we illustrate the evolution of a small perturbation to the initial condition which generated Figure 2.7a; to be explicit we plot the evolution of the error in the Euclidean norm $|\cdot|$, for an initial perturbation of magnitude 10^{-4}. Figure 2.6 suggests that the measure μ_∞ is supported on a strange set with Lebesgue measure zero, and this is indeed the case; for this example there is no Lebesgue density ρ_∞ for the invariant measure, reflecting the fact that the attractor has a fractal dimension less than three, the dimension of the space where the dynamical system lies.

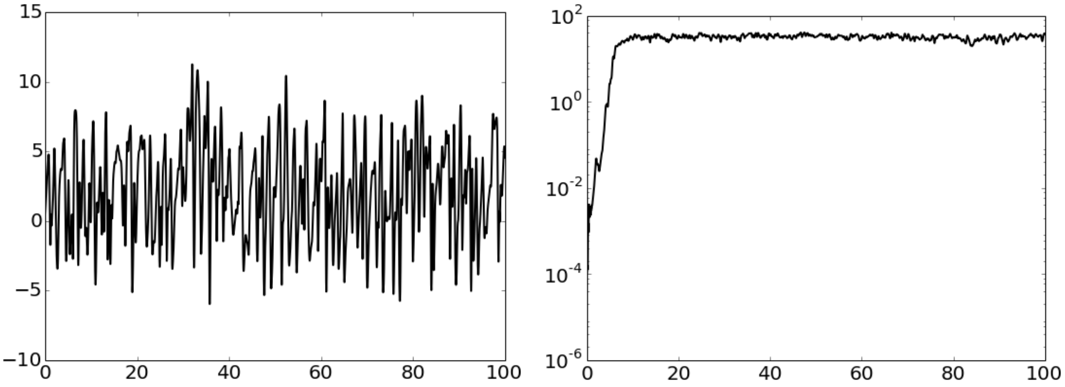
[2] Here index denotes components of the solution, not discrete time.



(a) u_1 as a function of time

(b) Evolution of error for a small perturbation

Figure 2.7: Dynamics of the Lorenz'63 model in the chaotic regime $(a, b, r) = (10, \frac{8}{3}, 28)$



(a) v_1 as a function of time

(b) Evolution of error for a small initial perturbation

Figure 2.8: Dynamics of the Lorenz'96 model in the chaotic regime $(F, K) = (8, 40)$

♠

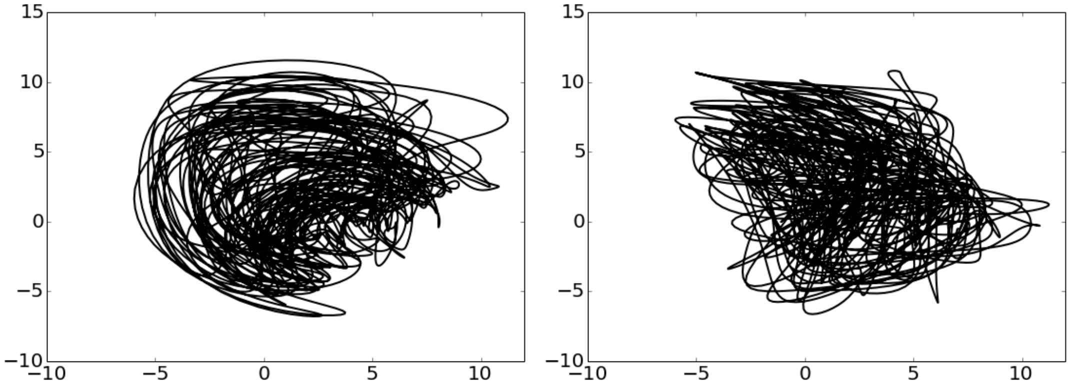
Example 2.7 *The Lorenz '96 model is a simple dynamical system, of tunable dimension, which was designed as a caricature of the dynamics of Rossby waves in atmospheric dynamics. The equations have a periodic "ring" formulation and take the form*[3]

$$
\begin{aligned}
\frac{dv_k}{dt} &= v_{k-1}\big(v_{k+1} - v_{k-2}\big) - v_k + F, \quad k \in \{1, \cdots, K\}, && (2.18a)\\
v_0 &= v_K, \quad v_{K+1} = v_1, \quad v_{-1} = v_{K-1}. && (2.18b)
\end{aligned}
$$

Equation (2.18) *satisfies the same dissipativity property* (2.16) *satisfied by the Lorenz '63 model, for appropriate choice of $\alpha, \beta > 0$, and hence also satisfies the absorbing ball property* (2.17) *thus having a global attractor (see Definition 1.23).*

In Figure 2.8a we plot a trajectory of v_1 versus time for $F = 8$ and $K = 40$. Furthermore, as we did in the case of the Lorenz '63 model, we also show the evolution of the Euclidean

[3] Again, here index denotes components of the solution, not discrete time.



(a) v_1 vs v_K

(b) v_1 vs v_{K-1}

Figure 2.9: Projection of the Lorenz'96 attractor onto two different pairs of coordinates.

norm of the error $|\cdot|$ *for an initial perturbation of magnitude* 10^{-4}*; this is displayed in Figure 2.8b and clearly demonstrates sensitive dependence on initial conditions. We visualize the attractor, projected onto two different pairs of coordinates, in Figure 2.9.*

♠

2.3 Smoothing Problem

2.3.1. Probabilistic Formulation of Data Assimilation

Together (2.1) and (2.2) provide a probabilistic model for the jointly varying random variable (v, y). In the case of deterministic dynamics, (2.3) and (2.2) provide a probabilistic model for the jointly varying random variable (v_0, y). Thus in both cases we have a random variable (u, y), with $u = v$ (resp. $u = v_0$) in the stochastic (resp. deterministic) case. Our aim is to find out information about the signal v, in the stochastic case, or v_0 in the deterministic case, from observation of a single instance of the data y. The natural probabilistic approach to this problem is to try and find the probability measure describing the random variable u given y, denoted $u|y$. This constitutes the Bayesian formulation of the problem of determining information about the signal arising in a noisy dynamical model, based on noisy observations of that signal. We will refer to the conditioned random variable $u|y$, in the case of either the stochastic dynamics or deterministic dynamics, as the **smoothing distribution**. It is a random variable which contains all the probabilistic information about the signal, given our observations. The key concept which drives this approach is Bayes' formula from subsection 1.1.4 which we use repeatedly in what follows.

2.3.2. Stochastic Dynamics

We wish to find the signal v from (2.1) from a single instance of data y given by (2.2). To be more precise we wish to condition the signal on a discrete time interval $\mathbb{J}_0 = \{0, ..., J\}$, given data on the discrete time interval $\mathbb{J} = \{1, ..., J\}$; we refer to $\mathbb{J}_0$ as the data assimilation window. We define $v = \{v_j\}_{j\in\mathbb{J}_0}, y = \{y_j\}_{j\in\mathbb{J}}, \xi = \{\xi_j\}_{j\in\mathbb{J}_0}$ and $\eta = \{\eta_j\}_{j\in\mathbb{J}}$. The smoothing distribution here is the distribution of the conditioned random variable $v|y$. Recall that we have assumed that v_0, ξ and η are mutually independent random variables. With this fact in hand we may apply Bayes' formula to find the pdf $\mathbb{P}(v|y)$.

Prior The prior on v is specified by (2.1), together with the independence of u and ξ and the i.i.d. structure of ξ. First note that, using (1.5) and the i.i.d. structure of ξ in turn, we obtain

$$\begin{aligned}\mathbb{P}(v) &= \mathbb{P}(v_J, v_{J-1}, \cdots, v_0)\\ &= \mathbb{P}(v_J|v_{J-1}, \cdots, v_0)\mathbb{P}(v_{J-1}, \cdots, v_0)\\ &= \mathbb{P}(v_J|v_{J-1})\mathbb{P}(v_{J-1}, \cdots, v_0).\end{aligned}$$

Proceeding inductively gives

$$\mathbb{P}(v) = \prod_{j=0}^{J-1} \mathbb{P}(v_{j+1}|v_j)\mathbb{P}(v_0).$$

Now

$$\mathbb{P}(v_0) \propto \exp\Bigl(-\frac{1}{2}\bigl|C_0^{-\frac{1}{2}}(v_0 - m_0)\bigr|^2\Bigr)$$

whilst

$$\mathbb{P}(v_{j+1}|v_j) \propto \exp\Bigl(-\frac{1}{2}\Bigl|\Sigma^{-\frac{1}{2}}\bigl(v_{j+1} - \Psi(v_j)\bigr)\Bigr|^2\Bigr).$$

The probability distribution $\mathbb{P}(v)$ that we now write down is *not* Gaussian, but the distribution on the initial condition $\mathbb{P}(v_0)$, and the conditional distributions $\mathbb{P}(v_{j+1}|v_j)$, are all Gaussian, making the explicit calculations above straightforward.

Combining the preceding information we obtain

$$\mathbb{P}(v) \propto \exp(-\mathsf{J}(v))$$

where

$$\begin{aligned}\mathsf{J}(v) :=& \quad \tfrac{1}{2}\bigl|C_0^{-\frac{1}{2}}(v_0 - m_0)\bigr|^2 + \textstyle\sum_{j=0}^{J-1} \tfrac{1}{2}\bigl|\Sigma^{-\frac{1}{2}}\bigl(v_{j+1} - \Psi(v_j)\bigr)\bigr|^2 & (2.19\text{a})\\ =& \quad \tfrac{1}{2}\bigl|v_0 - m_0\bigr|_{C_0}^2 + \textstyle\sum_{j=0}^{J-1} \tfrac{1}{2}\bigl|v_{j+1} - \Psi(v_j)\bigr|_{\Sigma}^2. & (2.19\text{b})\end{aligned}$$

The pdf $\mathbb{P}(v) = \rho_0(v)$ proportional to $\exp(-\mathsf{J}(v))$ determines a prior measure μ_0 on $\mathbb{R}^{|\mathbb{J}_0|\times n}$. The fact that the probability is not, in general, Gaussian follows from the fact that Ψ is not, in general, linear.

Likelihood The likelihood of the data $y|v$ is determined as follows. It is a (Gaussian) probability distribution on $\mathbb{R}^{|\mathbb{J}|\times m}$, with pdf $\mathbb{P}(y|v)$ proportional to $\exp(-\Phi(v;y))$, where

$$\Phi(v;y) = \sum_{j=0}^{J-1} \frac{1}{2}\bigl|y_{j+1} - h(v_{j+1})\bigr|_{\Gamma}^2. \tag{2.20}$$

To see this note that, because of the i.i.d. nature of the sequence η, it follows that

$$\begin{aligned}\mathbb{P}(y|v) &= \prod_{j=0}^{J-1} \mathbb{P}(y_{j+1}|v)\\ &= \prod_{j=0}^{J-1} \mathbb{P}(y_{j+1}|v_{j+1})\\ &\propto \prod_{j=0}^{J-1} \exp\Bigl(-\frac{1}{2}\bigl|\Gamma^{-\frac{1}{2}}\bigl(y_{j+1} - h(v_{j+1})\bigr)\bigr|^2\Bigr)\\ &= \exp(-\Phi(v;y)).\end{aligned}$$

In the applied literature m_0 and C_0 are often referred to as the **background mean** and **background covariance** respectively; we refer to Φ as the **model-data misfit** functional.

Using Bayes' formula (1.7) we can combine the prior and the likelihood to determine the posterior distribution, that is the smoothing distribution, on $v|y$. We denote the measure with this distribution by μ.

Theorem 2.8 *The posterior smoothing distribution on $v|y$ for the stochastic dynamics model* (2.1), (2.2) *is a probability measure μ on $\mathbb{R}^{|\mathbb{J}_0|\times n}$ with pdf $\mathbb{P}(v|y) = \rho(v)$ proportional to* $\exp(-\mathsf{I}(v;y))$ *where*

$$\mathsf{I}(v;y) = \mathsf{J}(v) + \Phi(v;y). \tag{2.21}$$

Proof Bayes' formula (1.7) gives us

$$\mathbb{P}(v|y) = \frac{\mathbb{P}(y|v)\mathbb{P}(v)}{\mathbb{P}(y)}.$$

Thus, ignoring constants of proportionality which depend only on y,

$$\begin{aligned}\mathbb{P}(v|y) &\propto \mathbb{P}(y|v)\mathbb{P}(v_0)\\ &\propto \exp(-\Phi(v;y))\exp(-\mathsf{J}(v))\\ &= \exp(-\mathsf{I}(v;y)).\end{aligned}$$

□

Note that, although the preceding calculations required only knowledge of the pdfs of Gaussian distributions, the resulting posterior distribution is non-Gaussian in general, unless Ψ and h are linear. This is because, unless Ψ and h are linear, $\mathsf{I}(\cdot;y)$ is not quadratic. We refer to I as the negative log-posterior. It will be helpful later to note that

$$\frac{\rho(v)}{\rho_0(v)} \propto \exp\bigl(-\Phi(v;y)\bigr). \tag{2.22}$$

2.3.3. Reformulation of Stochastic Dynamics

For the development of algorithms to probe the posterior distribution, the following reformulation of the stochastic dynamics problem can be very useful. For this we define the vector $\xi = (v_0, \xi_0, \xi_1, \cdots, \xi_{J-1}) \in \mathbb{R}^{|\mathbb{J}_0|n}$. The following lemma is key to what follows.

Lemma 2.9 *Define the mapping $G : \mathbb{R}^{|\mathbb{J}_0|\times n} \mapsto \mathbb{R}^{|\mathbb{J}_0|\times n}$ by*

$$G_j(v_0, \xi_0, \xi_1, \cdots, \xi_{J-1}) = v_j, \quad j = 0, \cdots, J,$$

where v_j is determined by (2.1)*. Then this mapping is invertible. Furthermore, if $\Psi \equiv 0$, then G is the identity mapping.*

Proof In words the mapping G takes the initial condition and noise into the signal. Invertibility requires determination of the initial condition and the noise from the signal. From the signal we may compute the noise as follows noting that, of course, the initial condition is specified and that then we have

$$\xi_j = v_{j+1} - \Psi(v_j), \quad j = 0, \cdots, J-1.$$

The fact that G becomes the identity mapping when $\Psi \equiv 0$ follows directly from (2.1) by inspection. □

We may thus consider the smoothing problem as finding the probability distribution of ξ, as defined prior to the lemma, given data y, with y as defined in section 2.3.2. Furthermore we have, using the notion of pushforward,

$$\mathbb{P}(v|y) = G \star \mathbb{P}(\xi|y), \quad \mathbb{P}(\xi|y) = G^{-1} \star \mathbb{P}(v|y). \tag{2.23}$$

These formulae mean that it is easy to move between the two measures: samples from one can be converted into samples from the other simply by applying G or G^{-1}. This means that algorithms can be applied to, for example, generate samples from $\xi|y$, and then convert into samples from $v|y$. We will use this later on. In order to use this idea it will be helpful to have an explicit expression for the pdf of $\xi|y$. We now find such an expression.

To start we introduce the measure ϑ_0 with density π_0 found from μ_0 and ρ_0 in the case where $\Psi \equiv 0$. Thus

$$\pi_0(v) \quad \propto \quad \exp\left(-\frac{1}{2}\left|C_0^{-\frac{1}{2}}(v_0 - m_0)\right|^2 - \sum_{j=0}^{J-1} \frac{1}{2}|\Sigma^{-\frac{1}{2}} v_{j+1}|^2\right) \tag{2.24a}$$

$$\propto \quad \exp\left(-\frac{1}{2}\left|v_0 - m_0\right|_{C_0}^2 - \sum_{j=0}^{J-1} \frac{1}{2}|v_{j+1}|_{\Sigma}^2\right) \tag{2.24b}$$

and hence ϑ_0 is a Gaussian measure, independent in each component v_j for $j = 0, \cdots, J$. By Lemma 2.9 we also deduce that measure ϑ_0 with density π_0 is the prior on ξ as defined above:

$$\pi_0(\xi) \propto \exp\left(-\frac{1}{2}\left|v_0 - m_0\right|_{C_0}^2 - \sum_{j=0}^{J-1} \frac{1}{2}|\xi_j|_{\Sigma}^2\right). \tag{2.25}$$

We now compute the likelihood of $y|\xi$. For this we define

$$\mathcal{G}_j(\xi) = h\bigl(G_j(\xi)\bigr) \tag{2.26}$$

and note that we may then concatenate the data and write

$$y = \mathcal{G}(\xi) + \eta \tag{2.27}$$

where $\eta = (\eta_1, \cdots, \eta_J)$ is a the Gaussian random variable $N(0, \Gamma_J)$ where Γ_J is a block diagonal $nJ \times nJ$ matrix with $n \times n$ diagonal blocks Γ. It follows that the likelihood is determined by $\mathbb{P}(y|\xi) = N\bigl(\mathcal{G}(\xi), \Gamma_J\bigr)$. Applying Bayes formula from (1.7) to find the pdf for $\xi|y$ we find the posterior ϑ on $\xi|y$, as summarized in the following theorem.

Theorem 2.10 *The posterior smoothing distribution on $\xi|y$ for the stochastic dynamics model* (2.1), (2.2) *is a probability measure ϑ on $\mathbb{R}^{|\mathbb{J}_0|\times n}$ with pdf $\mathbb{P}(\xi|y) = \pi(\xi)$ proportional to* $\exp(-\mathsf{I}_r(\xi; y))$ *where*

$$\mathsf{I}_r(\xi; y) = \mathsf{J}_r(\xi) + \Phi_r(\xi; y), \tag{2.28}$$

$$\Phi_r(\xi; y) := \frac{1}{2}\left|(y - \mathcal{G}(\xi))\right|_{\Gamma_J}^2$$

and

$$\mathsf{J}_r(\xi) := \frac{1}{2}\left|v_0 - m_0\right|_{C_0}^2 + \sum_{j=0}^{J-1} \frac{1}{2}\left|\xi_j\right|_{\Sigma}^2.$$

We refer to I_{r} as the negative log-posterior.

2.3.4. Deterministic Dynamics

It is also of interest to study the posterior distribution on the initial condition in the case where the model dynamics contains no noise, and is given by (2.3); this we now do. Recall that $\Psi^{(j)}(\cdot)$ denotes the $j-$fold composition of $\Psi(\cdot)$ with itself. In the following we sometimes refer to $\mathsf{J}_{\rm det}$ as the **background** penalization, and m_0 and C_0 as the background mean and covariance; we refer to $\Phi_{\rm det}$ as the **model-data misfit** functional.

Theorem 2.11 *The posterior smoothing distribution on* $v_0|y$ *for the the deterministic dynamics model* (2.3), (2.2) *is a probability measure* ν *on* $\mathbb{R}^n$ *with density* $\mathbb{P}(v_0|y) = \varrho(v_0)$ *proportional to* $\exp(-\mathsf{I}_{\rm det}(v_0;y))$ *where*

$$\mathsf{I}_{\rm det}(v_0;y) = \mathsf{J}_{\rm det}(v_0) + \Phi_{\rm det}(v_0;y), \tag{2.29a}$$

$$\mathsf{J}_{\rm det}(v_0) = \frac{1}{2}\big|v_0 - m_0\big|_{C_0}^2, \tag{2.29b}$$

$$\Phi_{\rm det}(v_0;y) = \sum_{j=0}^{J-1}\frac{1}{2}\big|y_{j+1} - h\big(\Psi^{(j+1)}(v_0)\big)\big|_{\Gamma}^2. \tag{2.29c}$$

Proof We again use Bayes' rule which states that

$$\mathbb{P}(v_0|y) = \frac{\mathbb{P}(y|v_0)\mathbb{P}(v_0)}{\mathbb{P}(y)}.$$

Thus, ignoring constants of proportionality which depend only on y,

$$\begin{aligned}\mathbb{P}(v_0|y) &\propto \mathbb{P}(y|v_0)\mathbb{P}(v_0)\\ &\propto \exp\big(-\Phi_{\rm det}(v_0;y)\big)\exp\big(-\frac{1}{2}|v_0 - m_0|_{C_0}^2\big)\\ &= \exp(-\mathsf{I}_{\rm det}(v_0;y)).\end{aligned}$$

Here we have used the fact that $\mathbb{P}(y|v_0)$ is proportional to $\exp\big(-\Phi_{\rm det}(v_0;y)\big)$; this follows from the fact that $y_j|v_0$ form an i.i.d sequence of Gaussian random variables $N\big(h(v_j),\Gamma\big)$ with $v_j = \Psi^{(j)}(v_0)$. □

We refer to $\mathsf{I}_{\rm det}$ as the negative log-posterior.

2.4 Filtering Problem

The smoothing problem considered in the previous section involves, potentially, conditioning v_j on data y_k with $k > j$. Such conditioning can only be performed *off-line* and is of no use in *on-line* scenarios where we want to determine information on the state of the signal *now* hence using only data from the past up to the present. To study this situation, let $Y_j = \{y_l\}_{l=1}^j$ denote the accumulated data up to time j. **Filtering** is concerned with determining $\mathbb{P}(v_j|Y_j)$, the pdf associated with the probability measure on the random variable $v_j|Y_j$; in particular filtering is concerned with the sequential updating this pdf as the index j is incremented. This update is defined by the following procedure which provides a prescription for computing $\mathbb{P}(v_{j+1}|Y_{j+1})$ from $\mathbb{P}(v_j|Y_j)$ via two steps: **prediction** which computes the mapping $\mathbb{P}(v_j|Y_j) \mapsto \mathbb{P}(v_{j+1}|Y_j)$ and **analysis** which computes $\mathbb{P}(v_{j+1}|Y_j) \mapsto \mathbb{P}(v_{j+1}|Y_{j+1})$ by application of Bayes' formula.

Prediction Note that $\mathbb{P}(v_{j+1}|Y_j, v_j) = \mathbb{P}(v_{j+1}|v_j)$ because Y_j contains noisy and indirect information about v_j and cannot improve upon perfect knowledge of the variable v_j. Thus, by (1.5), we deduce that

$$\mathbb{P}(v_{j+1}|Y_j) = \int_{\mathbb{R}^n} \mathbb{P}(v_{j+1}|Y_j, v_j)\mathbb{P}(v_j|Y_j)dv_j \tag{2.30a}$$

$$= \int_{\mathbb{R}^n} \mathbb{P}(v_{j+1}|v_j)\mathbb{P}(v_j|Y_j)dv_j \tag{2.30b}$$

Note that, since the forward model equation (2.1) determines $\mathbb{P}(v_{j+1}|v_j)$, this prediction step provides the map from $\mathbb{P}(v_j|Y_j)$ to $\mathbb{P}(v_{j+1}|Y_j)$. This prediction step simplifies in the case of deterministic dynamics (2.3); in this case it simply corresponds to computing the pushforward of $\mathbb{P}(v_j|Y_j)$ under the map Ψ.

Analysis Note that $\mathbb{P}(y_{j+1}|v_{j+1}, Y_j) = \mathbb{P}(y_{j+1}|v_{j+1})$ because Y_j contains noisy and indirect information about v_j and cannot improve upon perfect knowledge of the variable v_{j+1}. Thus, using Bayes' formula (1.7), we deduce that

$$\begin{aligned}\mathbb{P}(v_{j+1}|Y_{j+1}) &= \mathbb{P}(v_{j+1}|Y_j, y_{j+1})\\ &= \frac{\mathbb{P}(y_{j+1}|v_{j+1}, Y_j)\mathbb{P}(v_{j+1}|Y_j)}{\mathbb{P}(y_{j+1}|Y_j)}\\ &= \frac{\mathbb{P}(y_{j+1}|v_{j+1})\mathbb{P}(v_{j+1}|Y_j)}{\mathbb{P}(y_{j+1}|Y_j)}.\end{aligned} \tag{2.31}$$

Since the observation equation (2.2) determines $\mathbb{P}(y_{j+1}|v_{j+1})$, this analysis step provides a map from $\mathbb{P}(v_{j+1}|Y_j)$ to $\mathbb{P}(v_{j+1}|Y_{j+1})$.

Filtering Update Together, then, the prediction and analysis step provide a mapping from $\mathbb{P}(v_j|Y_j)$ to $\mathbb{P}(v_{j+1}|Y_{j+1})$. Indeed if we let μ_j denote the probability measure on $\mathbb{R}^n$ corresponding to the density $\mathbb{P}(v_j|Y_j)$ and $\widehat{\mu}_{j+1}$ be the probability measure on $\mathbb{R}^n$ corresponding to the density $\mathbb{P}(v_{j+1}|Y_j)$ then the prediction step maps μ_j to $\widehat{\mu}_{j+1}$ whilst the analysis step maps $\widehat{\mu}_{j+1}$ to μ_{j+1}. However there is, in general, no easily usable closed form expression for the density of μ_j, namely $\mathbb{P}(v_j|Y_j)$. Nevertheless, formulae (2.30), (2.31) form the starting point for numerous algorithms to approximate $\mathbb{P}(v_j|Y_j)$. In terms of analyzing the particle filter it is helpful conceptually to write the prediction and analysis steps as

$$\widehat{\mu}_{j+1} = P\mu_j \qquad \mu_{j+1} = L_j\widehat{\mu}_{j+1}. \tag{2.32}$$

Note that P does not depend on j as the same Markov process governs the prediction step at each j; however L_j depends on j because the likelihood sees different data at each j. Furthermore, the formula $\widehat{\mu}_{j+1} = P\mu_j$ summarizes (2.30) whilst $\mu_{j+1} = L_j\widehat{\mu}_{j+1}$ summarizes (2.31). Note that P is a linear mapping, whilst L_j is nonlinear; this issue is discussed in subsections 1.4.1 and 1.4.3, at the level of pdfs.

2.5 Filtering and Smoothing are Related

The filtering and smoothing approaches to determining the signal from the data are distinct, but related. They are related by the fact that in both cases the solution computed at the *end* of any specified time-interval is conditioned on the same data, and must hence coincide; this is made precise in the following.

Theorem 2.12 *Let $\mathbb{P}(v|y)$ denote the smoothing distribution on the discrete time interval $j \in \mathbb{J}_0$, and $\mathbb{P}(v_J|Y_J)$ the filtering distribution at time $j = J$ for the stochastic dynamics model* (2.1). *Then the marginal of the smoothing distribution on v_J is the same as the filtering distribution at time J:*

$$\int \mathbb{P}(v|y) dv_0 dv_1 ... dv_{J-1} = \mathbb{P}(v_J|Y_J).$$

Proof Note that $y = Y_J$. Since $v = (v_0, ..., v_{J-1}, v_J)$ the result follows trivially. □

Remark 2.13 *Note that the marginal of the smoothing distribution on say v_j, $j < J$ is* not *equal to the filter $\mathbb{P}(v_j|Y_j)$. This is because the smoother induces a distribution on v_j which is influenced by the entire data set $Y_J = y = \{y_l\}_{l \in \mathbb{J}}$; in contrast the filter at j involves only the data $Y_j = \{y_l\}_{l \in \{1,...,j\}}$.* ♠

It is also interesting to mention the relationship between filtering and smoothing in the case of noise-free dynamics. In this case the filtering distribution $\mathbb{P}(v_j|Y_j)$ is simply found as the pushforward of the smoothing distribution on $\mathbb{P}(v_0|Y_j)$ under $\Psi^{(j)}$, that is under j applications of Ψ.

Theorem 2.14 *Let $\mathbb{P}(v_0|y)$ denote the smoothing distribution on the discrete time interval $j \in \mathbb{J}_0$, and $\mathbb{P}(v_J|Y_J)$ the filtering distribution at time $j = J$ for the deterministic dynamics model* (2.3). *Then the pushforward of the smoothing distribution on v_0 under $\Psi^{(J)}$ is the same as the filtering distribution at time J:*

$$\Psi^{(J)} \star \mathbb{P}(v_0|Y_J) = \mathbb{P}(v_J|Y_J).$$

2.6 Well-Posedness

Well-posedness of a mathematical problem refers, generally, to the existence of a unique solution which depends continuously on the parameters defining the problem. We have shown, for both filtering and smoothing, how to construct a uniquely defined probabilistic solution to the problem of determining the signal given the data. In this setting it is natural to consider well-posedness with respect to the data itself. Thus we now investigate the continuous dependence of the probabilistic solution on the observed data; indeed we will show Lipschitz dependence. To this end we need probability metrics, as introduced in section 1.3.

As we do throughout the notes, we perform all calculations using the existence of everywhere positive Lebesgue densities for our measures. We let μ_0 denote the prior measure on v for the smoothing problem arising in stochastic dynamics, as defined by (2.1). Then μ and μ' denote the posterior measures resulting from two different instances of the data, y and y' respectively. Let ρ_0, ρ and ρ' denote the Lebesgue densities on μ_0, μ and μ' respectively. Then, for J and Φ as defined in (2.19) and (2.20),

$$\rho_0(v) = \frac{1}{Z_0} \exp(-\mathsf{J}(v)), \tag{2.33a}$$

$$\rho(v) = \frac{1}{Z} \exp(-\mathsf{J}(v) - \Phi(v; y)), \tag{2.33b}$$

$$\rho'(v) = \frac{1}{Z'} \exp(-\mathsf{J}(v) - \Phi(v; y')), \tag{2.33c}$$

where

$$Z_0 = \int \exp(-\mathsf{J}(v))dv, \tag{2.34a}$$

$$Z = \int \exp(-\mathsf{J}(v) - \Phi(v;y))dv, \tag{2.34b}$$

$$Z' = \int \exp(-\mathsf{J}(v) - \Phi(v;y'))dv. \tag{2.34c}$$

Here, and in the proofs that follow in this section, all integrals are over $\mathbb{R}^{|\mathbb{J}_0|\times n}$ (or, in the case of the deterministic dynamics model at the end of the section, over $\mathbb{R}^n$). Note that $|\mathbb{J}_0|$ is the cardinality of the set $\mathbb{J}_0$ and is hence equal to $J+1$. To this end we note explicitly that (2.33a) implies that

$$\exp\bigl(-\mathsf{J}(v)\bigr)dv = Z_0\rho_0(v)dv = Z_0\mu_0(dv), \tag{2.35}$$

indicating that integrals weighted by $\exp\bigl(-\mathsf{J}(v)\bigr)$ may be rewritten as expectations with respect to μ_0. We use the identities (2.33), (2.34) and (2.35) repeatedly in what follows to express all integrals as expectations with respect to the measure μ_0. In particular the assumptions that we make for the subsequent theorems and corollaries in this section are all expressed in terms of expectations under μ_0 (or, under ν_0 for the deterministic dynamics problem considered at the end of the section). This is convenient because it relates to the unconditioned problem of stochastic dynamics for v, in the absence of any data, and may thus be checked once and for all, independently of the particular data set y or y' which are used to condition v and obtain μ and μ'.

We assume throughout what follows that y, y' are both contained in a ball of radius r in the Euclidean norm on $\mathbb{R}^{|\mathbb{J}|\times n}$. Again $|\mathbb{J}|$ is the cardinality of the set $\mathbb{J}$ and is hence equal to J. We also note that Z_0 is bounded from above independently of r, because ρ_0 is the density associated with the probability measure μ_0, which is therefore normalizable, and this measure is independent of the data. It also follows that $Z \le Z_0$, $Z' \le Z_0$ by using (2.35) in (2.34b), (2.34c), together with the fact that $\Phi(\cdot;y)$ is a positive function. Furthermore, if we assume that

$$\mathsf{v} := \sum_{j\in\mathbb{J}} \bigl(1 + |h(v_j)|^2\bigr) \tag{2.36}$$

satisfies $\mathbb{E}^{\mu_0}\mathsf{v} < \infty$, then both Z and Z' are positive with common lower bound depending only on r, as we now demonstrate. It is sufficient to prove the result for Z, which we now do. In the following, and in the proofs which follow, K denotes a generic constant, which may depend on r and J but not on the solution sequence v, and which may change from instance to instance. Note first that, by (2.34), (2.35),

$$\frac{Z}{Z_0} = \int \exp\bigl(-\Phi(v;y)\bigr)\rho_0(v)dv \ge \int \exp\bigl(-K\mathsf{v}\bigr)\rho_0(v)dv.$$

Since $\mathbb{E}^{\mu_0}\mathsf{v} < \infty$ we deduce from (1.2), that for R sufficiently large,

$$\begin{aligned}\frac{Z}{Z_0} \ge \exp\bigl(-KR\bigr)\int_{|\mathsf{v}|<R} \rho_0(v)dv &= \exp\bigl(-KR\bigr)\mathbb{P}^{\mu_0}(|\mathsf{v}| < R)\\ &\ge \exp\bigl(-KR\bigr)\bigl(1 - R^{-1}\mathbb{E}^{\mu_0}\mathsf{v}\bigr).\end{aligned}$$

Since K depends on y, y' only through r, we deduce that, by choice of R sufficiently large, we have found lower bounds on Z, Z' which depend on y, y' only through r.

Finally we note that, since all norms are equivalent on finite dimensional spaces, there is constant K, such that

$$\left(\sum_{j=0}^{J-1}|y_{j+1}-y'_{j+1}|_{\Gamma}^{2}\right)^{\frac{1}{2}}\leq K|y-y'|. \tag{2.37}$$

The following theorem then shows that the posterior measure is in fact Lipschitz continuous, in the Hellinger metric, with respect to the data.

Theorem 2.15 *Consider the smoothing problem arising from the stochastic dynamics model* (2.1)*, resulting in the posterior probability distributions μ and μ' associated with two different data sets y and y'. Assume that $\mathbb{E}^{\mu_0}\mathsf{v}<\infty$ where v is given by* (2.36)*. Then there exists $c=c(r)$ such that, for all $|y|,|y'|\leq r$,*

$$d_{\mathrm{Hell}}(\mu,\mu')\leq c|y-y'|.$$

Proof We have, by (2.33), (2.35),

$$\begin{aligned}
d_{\mathrm{Hell}}(\mu,\mu')^2&=\frac{1}{2}\int|\sqrt{\rho(v)}-\sqrt{\rho'(v)}|^2dv\\
&=\frac{1}{2}\int Z_0\Big|\frac{1}{\sqrt{Z}}e^{-\frac{1}{2}\Phi(v;y)}-\frac{1}{\sqrt{Z'}}e^{-\frac{1}{2}\Phi(v;y')}\Big|^2\rho_0(v)dv\\
&\leq I_1+I_2,
\end{aligned}$$

where

$$I_1=Z_0\int\frac{1}{Z}\Big|e^{-\frac{1}{2}\Phi(v;y)}-e^{-\frac{1}{2}\Phi(v;y')}\Big|^2\rho_0(v)dv$$

and, using (2.33) and (2.34),

$$\begin{aligned}
I_2&=Z_0\Big|\frac{1}{\sqrt{Z}}-\frac{1}{\sqrt{Z'}}\Big|^2\int e^{-\Phi(v;y')}\rho_0(v)dv\\
&=Z'\Big|\frac{1}{\sqrt{Z}}-\frac{1}{\sqrt{Z'}}\Big|^2.
\end{aligned}$$

We estimate I_2 first. Since, as shown before the theorem, Z,Z' are bounded below by a positive constant depending only on r, we have

$$I_2=\frac{1}{Z}|\sqrt{Z}-\sqrt{Z'}|^2=\frac{1}{Z}\frac{|Z-Z'|^2}{|\sqrt{Z}+\sqrt{Z'}|^2}\leq K|Z-Z'|^2.$$

As $\Phi(v;y)\geq 0$ and $\Phi(v;y')\geq 0$ we have from (2.33), (2.34), using the fact that e^{-x} is Lipschitz on $\mathbb{R}^+$,

$$\begin{aligned}
|Z-Z'|&\leq Z_0\int|e^{-\Phi(v;y)}-e^{-\Phi(v;y')}|\rho_0(v)dv\\
&\leq Z_0\int|\Phi(v;y)-\Phi(v;y')|\rho_0(v)dv.
\end{aligned}$$

By definition of Φ and use of (2.37)

$$\begin{aligned}
|\Phi(v;y)-\Phi(v;y')| &\le \frac{1}{2}\sum_{j=0}^{J-1}|y_{j+1}-y'_{j+1}|_\Gamma|y_{j+1}+y'_{j+1}-2h(v_{j+1})|_\Gamma\\
&\le \frac{1}{2}\left(\sum_{j=0}^{J-1}|y_{j+1}-y'_{j+1}|_\Gamma^2\right)^{\frac{1}{2}}\left(\sum_{j=0}^{J-1}|y_{j+1}+y'_{j+1}-2h(v_{j+1})|_\Gamma^2\right)^{\frac{1}{2}}\\
&\le K|y-y'|\left(\sum_{j=0}^{J-1}\bigl(1+|h(v_{j+1})|^2\bigr)\right)^{\frac{1}{2}}\\
&= K|y-y'|\mathsf{v}^{\frac{1}{2}}.
\end{aligned}$$

Since $\mathbb{E}^{\mu_0}\mathsf{v}<\infty$ implies that $\mathbb{E}^{\mu_0}\mathsf{v}^{\frac{1}{2}}<\infty$ it follows that

$$|Z-Z'|\le K|y-y'|.$$

Hence $I_2\le K|y-y'|^2$

Now, using that Z_0 is bounded above independently of r, that Z is bounded below, depending on data only through r, and that $e^{-\frac{1}{2}x}$ is Lipschitz on $\mathbb{R}^+$, it follows that I_1 satisfies

$$I_1\le K\int|\Phi(v;y)-\Phi(v;y')|^2\rho_0(v)dv.$$

Squaring the preceding bound on $|\Phi(v;y)-\Phi(v;y')|$ gives

$$|\Phi(v;y)-\Phi(v;y')|^2\le K|y-y'|^2\mathsf{v}$$

and so $I_1\le K|y-y'|^2$ as required. □

Corollary 2.16 *Consider the smoothing problem arising from the stochastic dynamics model* (2.1)*, resulting in the posterior probability distributions μ and μ' associated with two different data sets y and y'. Assume that $\mathbb{E}^{\mu_0}\mathsf{v}<\infty$ where v is given by* (2.36)*. Let $f:\mathbb{R}^{|\mathbb{J}_0|\times n}\to\mathbb{R}^p$ be such that $\mathbb{E}^{\mu_0}|f(v)|^2<\infty$. Then there is $c=c(r)>0$ such that, for all $|y|,|y'|<r$,*

$$|\mathbb{E}^{\mu}f(v)-\mathbb{E}^{\mu'}f(v)|\le c|y-y'|.$$

Proof First note that, since $\Phi(v;y)\ge0$, Z_0 is bounded above independently of r, and since Z is bounded from below depending only on r, $\mathbb{E}^{\mu}|f(v)|^2\le c\mathbb{E}^{\mu_0}|f(v)|^2$; and a similar bound holds under μ'. The result follows from (1.13) and Theorem 2.15. □

Using the relationship between filtering and smoothing as described in the previous section, we may derive a corollary concerning the filtering distribution .

Corollary 2.17 *Consider the smoothing problem arising from the stochastic dynamics model* (2.1)*, resulting in the posterior probability distributions μ and μ' associated with two different data sets y and y'. Assume that $\mathbb{E}^{\mu_0}\mathsf{v}<\infty$ where v is given by* (2.36)*. Let $g:\mathbb{R}^n\to\mathbb{R}^p$ be such that $\mathbb{E}^{\mu_0}|g(v_J)|^2<\infty$. Then there is $c=c(r)>0$ such that, for all $|y|,|y'|<r$,*

$$|\mathbb{E}^{\mu_J}g(u)-\mathbb{E}^{\mu'_J}g(u)|\le c|Y_J-Y'_J|,$$

where μ_J and μ'_J denote the filtering distributions at time J corresponding to data Y_J,Y'_J respectively (i.e. the marginals of μ and μ' on the coordinate at time J).

Proof Since, by Theorem 2.12, μ_J is the marginal of the smoother on the v_J coordinate, the result follows from Corollary 2.16 by choosing $f(v) = g(v_J)$. □

A similar theorem, and corollaries, may be proved for the case of deterministic dynamics (2.3), and the posterior $\mathbb{P}(v_0|y)$. We state the theorem and leave its proof to the reader. We let ν_0 denote the prior Gaussian measure $N(m_0, C_0)$ on v_0 for the smoothing problem arising in deterministic dynamics, and ν and ν' the posterior measures on v_0 resulting from two different instances of the data, y and y' respectively. We also define

$$\mathsf{v}_0 := \sum_{j=0}^{J-1} \bigl(1 + \bigl|h\bigl(\Psi^{(j+1)}(v_0)\bigr)\bigr|^2\bigr).$$

Theorem 2.18 *Consider the smoothing problem arising from the deterministic dynamics model* (2.3)*. Assume that* $\mathbb{E}^{\nu_0}\mathsf{v}_0 < \infty$*. Then there is* $c = c(r) > 0$ *such that, for all* $|y|, |y'| \le r$*,*

$$d_{\text{Hell}}(\nu, \nu') \le c|y - y'|.$$

2.7 Assessing The Quality of Data Assimilation Algorithms

It is helpful when studying algorithms for data assimilation to ask two questions: (i) how informative is the data we have? (ii) how good is our algorithm at extracting this information? These are two separate questions, answers to both of which are required in the quest to understand how well we perform at extracting a signal, using model and data. We take the two questions in separately, in turn; however we caution that many applied papers entangle them both and simply measure algorithm quality by ability to reconstruct the signal.

Answering question (i) is independent of any particular algorithm: it concerns the properties of the Bayesian posterior pdf itself. In some cases we will be interested in studying the properties of the probability distribution on the signal, or the initial condition, for a particular instance of the data generated from a particular instance of the signal, which we call the **truth**. In this context we will use the notation $y^\dagger = \{y_j^\dagger\}$ to denote the *realization* of the data generated from a particular realization of the truth $v^\dagger = \{v_j^\dagger\}$. We first discuss properties of the smoothing problem for stochastic dynamics . **Posterior consistency** concerns the question of the limiting behaviour of $\mathbb{P}(v|y^\dagger)$ as either $J \to \infty$ (large data sets) or $|\Gamma| \to 0$ (small noise). A key question is whether $\mathbb{P}(v|y^\dagger)$ converges to the truth in either of these limits; this might happen, for example, if $\mathbb{P}(v|y^\dagger)$ becomes closer and closer to a Dirac probability measure centred on $v^\dagger$. When this occurs we say that the problem exhibits Bayesian posterior consistency; it is then of interest to study the rate at which the limit is attained. Such questions concern the information content of the data; they do not refer to any algorithm and therefore they are not concerned with the quality of any particular algorithm. When considering filtering, rather than smoothing, a particular instance of this question concerns marginal distributions: for example one may be concerned with posterior consistency of $\mathbb{P}(v_J|y_J^\dagger)$ with respect to a Dirac on $v_J^\dagger$ in the filtering case, see Theorem 2.12; for the case of deterministic dynamics the distribution $\mathbb{P}(v|y^\dagger)$ is completely determined by $\mathbb{P}(v_0|y^\dagger)$ (see Theorem 2.14) so one may discuss posterior consistency of $\mathbb{P}(v_0|y^\dagger)$ with respect to a Dirac on $v_0^\dagger$.

Here it is appropriate to mention the important concept of **model error**. In many (in fact most) applications the physical system which generates the data set $\{y_j\}$ can be (sometimes significantly) different from the mathematical model used, at least in certain aspects. This can be thought of conceptually by imagining data generated by (2.2), with $v^\dagger = \{v_j^\dagger\}$ governed by the deterministic dynamics

$$v_{j+1}^\dagger = \Psi_{\mathrm{true}}(v_j^\dagger),\ j \in \mathbb{Z}^+. \tag{2.38a}$$

$$v_0^\dagger = u \sim N(m_0, C_0). \tag{2.38b}$$

Here the function Ψ_{true} governs the dynamics of the truth which underlies the data. We assume that the true solution operator is not known to us exactly, and seek instead to combine the data with the stochastic dynamics model (2.1); the noise $\{\xi_j\}$ is used to allow for the discrepancy between the true solution operator Ψ_{true} and that used in our model, namely Ψ. It is possible to think of many variants on this situation. For example, the dynamics of the truth may be stochastic; or the dynamics of the truth may take place in a higher-dimensional space than that used in our models, and may need to be projected into the model space. Statisticians sometimes refer to the situation where the data source differs from the model used as **model misspecification**.

We now turn from the information content, or quality, of the data to the quality of algorithms for data assimilation. We discuss three approaches to assessing quality. The first fully Bayesian approach can be defined independently of the quality of the data. The second estimation approach entangles the properties of the algorithm with the quality of the data. We discuss these two approaches in the context of the smoothing problem for stochastic dynamics . The reader will easily see how to generalize to smoothing for deterministic dynamics, or to filtering. The third approach is widely used in operational numerical weather prediction and judges quality by the ability to predict.

Bayesian Quality Assessment. Here we assume that the algorithm under consideration provides an approximation $\mathbb{P}_{\mathrm{approx}}(v|y)$ to the true posterior distribution $\mathbb{P}(v|y)$. We ask the question: how close is $\mathbb{P}_{\mathrm{approx}}(v|y)$ to $\mathbb{P}(v|y)$. We might look for a distance measure between probability distributions, or we might simply compare some important moments of the distributions, such as the mean and covariance. Note that this version of quality assessment does not refer to the concept of a true solution $v^\dagger$. We may apply it with $y = y^\dagger$, but we may also apply it when there is model error present and the data comes from outside the model used to perform data assimilation . However, if combined with Bayesian posterior consistency, when $y = y^\dagger$, then the triangle inequality relates the output of the algorithm to the truth $v^\dagger$. Very few practitioners evaluate their algorithms by this measure. This reflects the fact that knowing the true distribution $\mathbb{P}(v|y)$ is often difficult in practical high dimensional problems. However it is arguably the case that practitioners should spend more time querying their algorithms from the perspective of Bayesian quality assessment since the algorithms are often used to make probabilistic statements and forecasts.

Signal Estimation Quality Assessment. Here we assume that the algorithm under consideration provides an approximation to the signal v underlying the data, which we denote by v_{approx}; thus v_{approx} attempts to determine and then *track* the true signal from the data. If the algorithm actually provides a probability distribution, then this estimate might be, for example, the mean. We ask the question: if the algorithm is applied in the situation where the data $y^\dagger$ is generated from the the signal $v^\dagger$, how close is v_{approx} to $v^\dagger$? There are two important effects at play here: the first is the information content of the data – does the data actually contain enough information to allow for accurate reconstruction of the signal in principle; and the second is the role of the specific algorithm used – does the specific algorithm in question have the ability to extract this information when it is present. This approach thus measures the overall effect of these two in combination.

Forecast Skill. In many cases the goal of data assimilation is to provide better forecasts of the future, for example in numerical weather prediction. In this context data assimilation algorithms can be benchmarked by their ability to make forecasts. This can be discussed in both the Bayesian quality and signal estimation senses. We first discuss Bayesian Estimation forecast skill in the context of stochastic dynamics . The Bayesian k-lag forecast skill can be defined by studying the distance between the approximation $\mathbb{P}_{\text{approx}}(v|y)$ and $\mathbb{P}(v|y)$ when both are pushed forward from the end-point of the data assimilation window by k applications of the dynamical model (2.1); this model defines a Markov transition kernel which is applied $k-$times to produce a forecast. We now discuss signal estimation forecast skill in the context of deterministic dynamics. Using v_{approx} at the end point of the assimilation window as an initial condition, we run the model (2.3) forward by k steps and compare the output with $v^{\dagger}_{j+k}$. In practical application, this forecast methodology inherently confronts the effect of model error , since the data used to test forecasts is real data which is not generated by the model used to assimilate, as well as information content in the data and algorithm quality.

2.8 Illustrations

In order to build intuition concerning the probabilistic viewpoint on data assimilation we describe some simple examples where the posterior distribution may be visualized easily. For this reason we concentrate on the case of one-dimensional deterministic dynamics; the posterior pdf $\mathbb{P}(v_0|y)$ for deterministic dynamics is given by Theorem 2.11. It is one-dimensional when the dynamics is one-dimensional and takes place in $\mathbb{R}$. In section 3 we will introduce more sophisticated sampling methods to probe probability distributions in higher dimensions which arise from noisy dynamics and/or from high dimensional models.

Figure 2.10 concerns the scalar linear problem from Example 2.1 (recall that throughout this section we consider only the case of deterministic dynamics) with $\lambda = 0.5$. We employ a prior $N(4, 5)$, we assume that $h(v) = v$, and we set $\Gamma = \gamma^2$ and consider two different values of γ and two different values of J, the number of observations. The figure shows the posterior distribution in these various parameter regimes. The true value of the initial condition which underlies the data is $v^{\dagger}_0 = 0.5$. For both $\gamma = 1.0$ and 0.1 we see that, as the number of observations J increases, the posterior distribution appears to converge to a limiting distribution. However for smaller γ the limiting distribution has much smaller variance, and is centred closer to the true initial condition at 0.5. Both of these observations can be explained, using the fact that the problem is explicitly solvable: we show that for fixed γ and $J \to \infty$ the posterior distribution has a limit, which is a Gaussian with non-zero variance. And for fixed J as $\gamma \to 0$ the posterior distribution converges to a Dirac measure (Gaussian with zero variance) centred at the truth $v^{\dagger}_0$.

To see these facts we start by noting that from Theorem 2.11 the posterior distribution on $v_0|y$ is proportional to the exponential of

$$\mathsf{I}_{\text{det}}(v_0; y) = \frac{1}{2\gamma^2}\sum_{j=0}^{J-1}|y_{j+1} - \lambda^{j+1}v_0|^2 + \frac{1}{2\sigma_0^2}|v_0 - m_0|^2$$

where σ_0^2 denotes the prior variance C_0. As a quadratic form in v_0 this defines a Gaussian posterior distribution and we may complete the square to find the posterior mean m and variance σ^2_{post}:

$$\frac{1}{\sigma^2_{\text{post}}} = \frac{1}{\gamma^2}\sum_{j=0}^{J-1}\lambda^{2(j+1)} + \frac{1}{\sigma_0^2} = \frac{1}{\gamma^2}\Big(\frac{\lambda^2 - \lambda^{2J+2}}{1-\lambda^2}\Big) + \frac{1}{\sigma_0^2}$$

and

$$\frac{1}{\sigma^2_{\text{post}}}m = \frac{1}{\gamma^2}\sum_{j=0}^{J-1}\lambda^{(j+1)}y_{j+1} + \frac{1}{\sigma_0^2}m_0.$$

We note immediately that the posterior variance is independent of the data. Furthermore, if we fix γ and let $J \to \infty$ then for any $|\lambda| < 1$ we see that the large J limit of the posterior variance is determined by

$$\frac{1}{\sigma^2_{\text{post}}} = \frac{1}{\gamma^2}\Big(\frac{\lambda^2}{1-\lambda^2}\Big) + \frac{1}{\sigma_0^2}$$

and is non-zero; thus uncertainty remains in the posterior, even in the limit of large data. On the other hand, if we fix J and let $\gamma \to 0$ then $\sigma^2_{\text{post}} \to 0$ so that uncertainty disappears in this limit. It is then natural to ask what happens to the mean. To this end we assume that the data is itself generated by the linear model of Example 2.1 so that

$$y_{j+1} = \lambda^{j+1}v_0^\dagger + \gamma\zeta_{j+1}$$

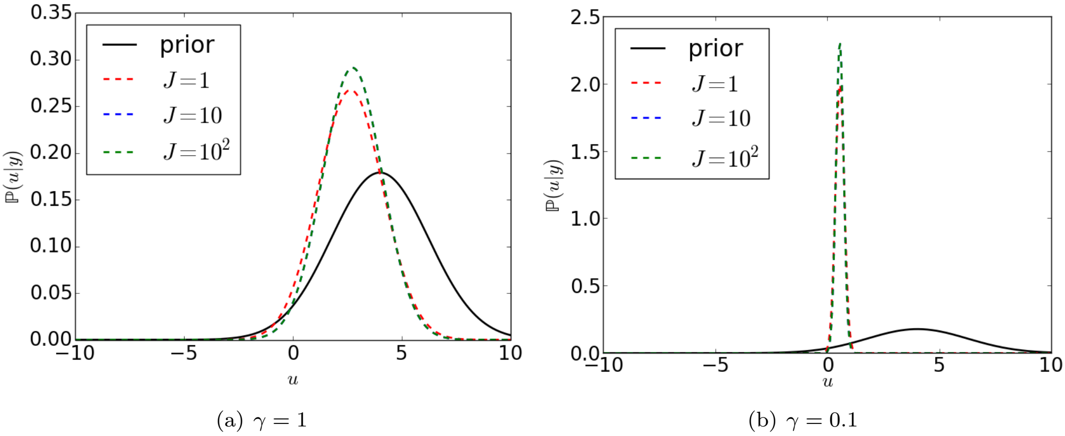
where ζ_j is an i.i.d. Gaussian sequence with $\zeta_1 \sim N(0,1)$. Then

$$\frac{1}{\sigma^2_{\text{post}}}m = \frac{1}{\gamma^2}\Big(\frac{\lambda^2-\lambda^{2J+2}}{1-\lambda^2}\Big)v_0^\dagger + \frac{1}{\gamma}\sum_{j=0}^{J-1}\lambda^{(j+1)}\zeta_{j+1} + \frac{1}{\sigma_0^2}m_0.$$

Using the formula for σ^2_{post} we obtain

$$\Big(\frac{\lambda^2-\lambda^{2J+2}}{1-\lambda^2}\Big)m + \frac{\gamma^2}{\sigma_0^2}m = \Big(\frac{\lambda^2-\lambda^{2J+2}}{1-\lambda^2}\Big)v_0^\dagger + \gamma\sum_{j=0}^{J-1}\lambda^{(j+1)}\zeta_{j+1} + \frac{\gamma^2}{\sigma_0^2}m_0.$$

From this it follows that, for fixed J and as $\gamma \to 0$, $m \to v_0^\dagger$, almost surely with respect to the noise realization $\{\zeta_j\}_{j\in\mathbb{J}}$. This is an example of posterior consistency.



(a) $\gamma = 1$

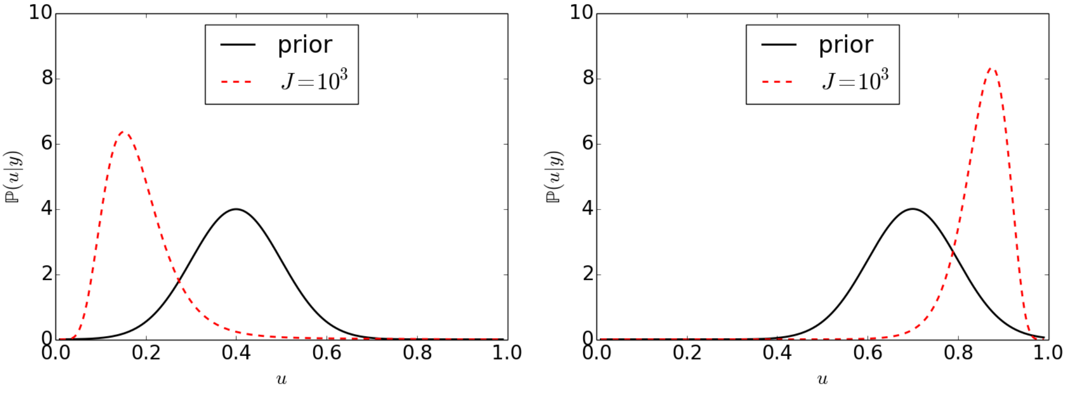
(b) $\gamma = 0.1$

Figure 2.10: Posterior distribution for Examples 2.1 for different levels of observational noise. The true initial condition used in both cases is $v_0 = 0.5$, while we have assumed that $C_0 = 5$ and $m_0 = 4$ for the prior distribution.

We now study Example 2.4 in which the true dynamics are no longer linear. We start our investigation taking $r = 2$ and investigate the effect of choosing different prior distributions.

Before discussing the properties of the posterior we draw attention to two facts. Firstly, as Figure 2.5a shows, the system converges in a small number of steps to the fixed point at 1/2 for this value of $r = 2$. And secondly the initial conditions v_0 and $1 - v_0$ both result in the same trajectory, if the initial condition is ignored. The first point implies that, after a small number of steps, the observed trajectory contains very little information about the initial condition. The second point means that, since we observe from the first step onwards, only the prior can distinguish between v_0 and $1 - v_0$ as the starting point.

Figure 2.11 concerns an experiment in which the true initial condition underlying the data is $v_0^\dagger = 0.1$. Two different priors are used, both with $C_0 = 0.01$, giving a standard deviation of 0.1, but with different means. The figure illustrates two facts: firstly, even with 10^3 observations, the posterior contains considerable uncertainty, reflecting the first point above. Secondly the prior mean has an important role in the form of the posterior pdf: shifting the prior mean to the right, from $m_0 = 0.4$ to $m_0 = 0.7$, results in a posterior which favours the initial condition $1 - v_0^\dagger$ rather than the truth $v_0^\dagger$.



(a) $r = 2$, $m_0 = 0.4$

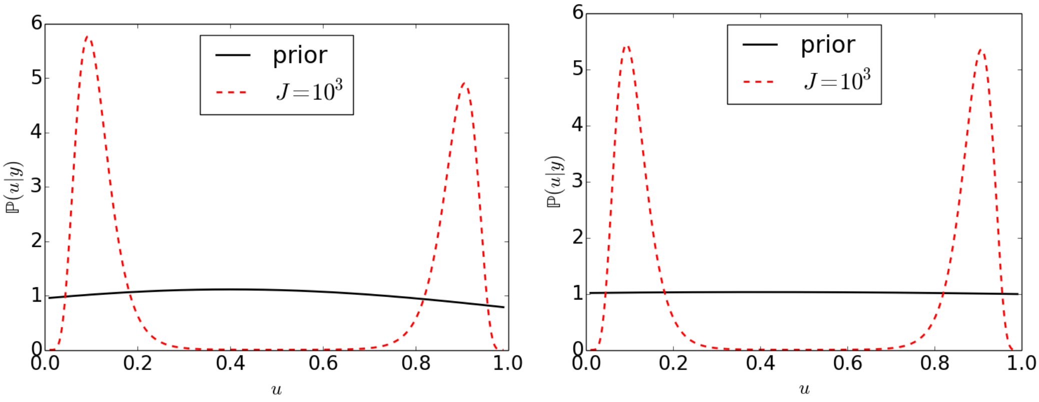
(b) $r = 2$ $m_0 = 0.7$

Figure 2.11: Posterior distribution for Example 2.4 for $r = 2$ in the case of different means for the prior distribution. We have used $C_0 = 0.01$, $\gamma = 0.1$ and true initial condition $v_0 = 0.1$, see also p2.m in section 5.1.2.

This behaviour of the posterior changes completely if we assume a flatter prior. This is illustrated in Figure 2.12 where we consider the prior $N(0.4, C_0)$ with $C_0 = 0.5$ and 5 respectively. As we increase the prior covariance the mean plays a much weaker role than in the preceding experiments: we now obtain a bimodal posterior centred around both the true initial condition $v_0^\dagger$, and also around $1 - v_0^\dagger$.

In Figure 2.13 we consider the quadratic map (2.9) with $r = 4$, $J = 5$ and prior $N(0.5, 0.01)$, with observational standard deviation $\gamma = 0.2$. Here, after only five observations the posterior is very peaked, although because of the $v \mapsto 1 - v$ symmetry mentioned above, there are two symmetrically related peaks; see Figure 2.13a. It is instructive to look at the negative of the logarithm of the posterior pdf which, upto an additive constant, is given by $\mathsf{I}_{\rm det}(v_0; y)$ in Theorem 2.11. The function $\mathsf{I}_{\rm det}(\cdot; y)$ is shown in Figure 2.13b. Its complexity indicates the considerable complications underlying solution of the smoothing problem. We will return to this last point in detail later. Here we simply observe that normalizing the posterior distribution requires evaluation of the integral

$$\int_{\mathbb{R}^n} e^{-\mathsf{I}(v_0, y)} dv_0.$$

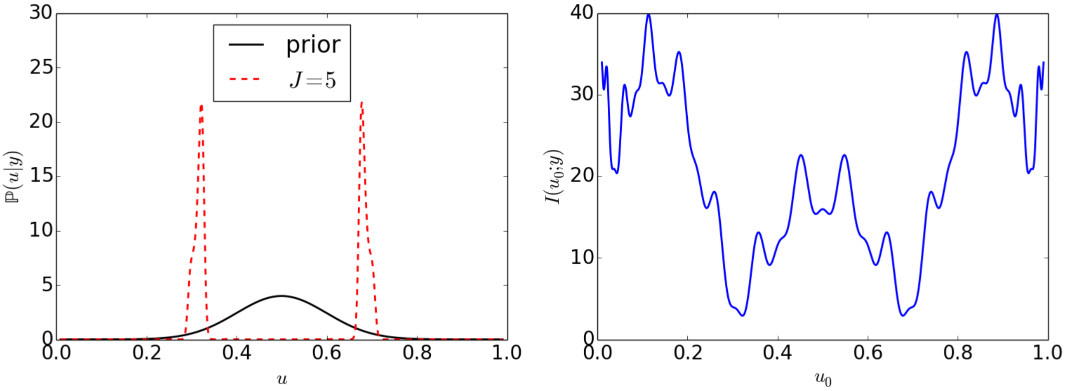


(a) $r = 2$, $C_0 = 0.5$

(b) $r = 2$ $C_0 = 5$

Figure 2.12: Posterior distribution for Example 2.4 for $r = 2$ in the case of different covariance for the prior distribution. We have used $m_0 = 0.4$, $\gamma = 0.1$ and true initial condition $v_0 = 0.1$.

This integral may often be determined almost entirely by very small subsets of $\mathbb{R}^n$, meaning that this calculation requires some care; indeed if $\mathsf{I}(\cdot)$ is very large over much of its domain then it may be impossible to compute the normalization constant numerically. We note, however, that the sampling methods that we will describe in the next chapter do not require evaluation of this integral.



(a) $r = 4$

(b) $\mathsf{I}(v_0; y)$ $r = 4, J = 5$

Figure 2.13: Posterior distribution and negative log posterior for Example 2.4 for $r = 4$ and $J = 5$. We have used $C_0 = 0.01, m_0 = 0.5$, $\gamma = 0.2$ and true initial condition $v_0 = 0.3$.

2.9 Bibliographic Notes

- Section 2.1 Data Assimilation has its roots in the geophysical sciences, and is driven by the desire to improve inaccurate models of complex dynamically evolving phenomena by means of incorporation of data. The book [66] describes data assimilation from the viewpoint of the atmospheric sciences and weather prediction, whilst the book [10] describes the subject from the viewpoint of oceanography. These two subjects were the initial drivers for evolution of the field. However, other applications are increasingly using the methodology of data assimilation, and the oil industry in particular is heavily involved in the use, and development, of algorithms in this area [93]. The recent book [1] provides a perspective on the subject from the viewpoint of physics and nonlinear dynamical systems, and includes motivational examples from neuroscience, as well as the geophysical sciences. The article [60] is a useful one to read because it establishes a notation which is now widely used in the applied communities and the articles [90, 6] provide simple introductions to various aspects of the subject from a mathematical perspective. The special edition of the journal PhysicaD, devoted to Data Assimilation, [61], provides an overview of the state of the art around a decade ago.

- It is useful to comment on generalizations of the set-up described in section 2.1. First we note that we have assumed a *Gaussian* structure for the additive noise appearing in both the signal model (2.1) and the data model (2.2). This is easily relaxed in much of what we describe here, provided that an explicit formula for the probability density function of the noise is known. However the Kalman filter, described in the next chapter, relies explicitly on the closed Gaussian form of the probability distributions resulting from the assumption of Gaussian noise. There are also other parts of the notes, such as the pCN MCMC methods and the minimization principle underlying approximate Gaussian filters, also both described in the next chapter, which require the Gaussian structure. Secondly we note that we have assumed *additive* noise. This, too, can be relaxed but has the complication that most non-additive noise models do not yield explicit formulae for the needed conditional probability density functions; for example this situation arises if one looks at stochastic differential equations over discrete time-intervals – see [15] and the discussion therein. However some of the methods we describe rely only on drawing samples from the desired distributions and do not require the explicit conditional probability density function . Finally we note that much of what we describe here translates to *infinite dimensional* spaces with respect to both the signal space, and the the data space; however in the infinite dimensional data space case the additive Gaussian observational noise is currently the only situation which is well-developed [106].

- Section 2.2. The subject of deterministic discrete time dynamical systems of the form (2.3) is overviewed in numerous texts; see [119] and the references therein, and Chapter 1 of [107], for example. The subject of stochastic discrete time dynamical systems of the form (2.1), and in particular the property of ergodicity which underlies Figure 2.4, is covered in some depth in [87]. The exact solutions of the quadratic map (2.9) for $r = 2$ and $r = 4$ may be found in [101] and [78] respectively. The Lorenz '63 model was introduced in [77]. Not only does this paper demonstrate the possibility of chaotic behaviour and sensitivity with respect to initial conditions, but it also makes a concrete connection between the three dimensional continuous time dynamical system and a one-dimensional chaotic map of the form (2.1). Furthermore, a subsequent computer assisted proof demonstrated rigorously that the ODE does indeed exhibit chaos [111, 112]. The book [105] discusses properties of the Lorenz '63 model in some detail and the book [41] discussed properties such as fractal dimension. The shift of origin that we have adopted for the Lorenz '63 model is explained in [109]; it enables the model to be written in an abstract form which includes many geophysical models of interest, such as the Lorenz '96 model introduced in [79], and the Navier-Stokes equation on a two-dimensional torus [84, 109]. We now briefly describe this common abstract form. The vector $u \in \mathbb{R}^J$ ($J = 3$ for Lorenz '63, J arbitrary for Lorenz' 96) solves the equation

$$\frac{du}{dt} + Au + B(u,u) = f, \quad u(0) = u_0, \tag{2.39}$$

where there is $\lambda > 0$ such that, for all $w \in \mathbb{R}^J$,

$$\langle Aw, w\rangle \ge \lambda |w|^2, \quad \langle B(w,w), w\rangle = 0.$$

Taking the inner-product with u shows that

$$\frac{1}{2}\frac{d}{dt}|u|^2 + \lambda |u|^2 \le \langle f, u\rangle.$$

If f is constant in time then this inequality may be used to show that (2.16) holds:

$$\frac{1}{2}\frac{d}{dt}|u|^2 \le \frac{1}{2\lambda}|f|^2 - \frac{\lambda}{2}|u|^2.$$

Integrating this inequality gives the existence of an absorbing set and hence leads to the existence of a global attractor; see Example 1.22, the book [109] or Chapter 2 of [107], for example.

- Section 2.3 contains the formulation of Data Assimilation as a fully nonlinear and non-Gaussian problem in Bayesian statistics. This formulation is not yet the basis of practical algorithms in the geophysical systems such as weather forecasting. This is because global weather forecast models involve $n = \mathcal{O}(10^9)$ unknowns, and incorporate $m = \mathcal{O}(10^6)$ data points daily; sampling the posterior on $\mathbb{R}^n$ given data in $\mathbb{R}^m$ in an online fashion, usable for forecasting, is beyond current algorithmic and computational capability. However the fully Bayesian perspective provides a fundamental mathematical underpinning of the subject, from which other more tractable approaches can be systematically derived. See [106] for discussion of the Bayesian approach to inverse problems. Historically, data assimilation has not evolved from this Bayesian perspective, but has rather evolved out of the control theory perspective. This perspective is summarized well in the book [63]. However, the importance of the Bayesian perspective is increasingly being recognized in the applied communities. In addition to providing a starting point from which to derive approximate algorithms, it also provides a gold standard against which other more *ad hoc* algorithms can be benchmarked; this use of Bayesian methodology was suggested in [71] in the context of meteorology (see discussion that follows), and then employed in [62] in the context of subsurface inverse problems arising in geophysics.

- Section 2.4 describes the filtering, or sequential, approach to data assimilation, within the fully Bayesian framework. For low dimensional systems the use of particle filters, which may be shown to rigorously approximate the required filtering distribution as it evolves in discrete time, has been enormously successful; see [39] for an overview. Unfortunately, these filters can behave poorly in high dimensions [17, 9, 103]. Whilst there is ongoing work to overcome these problems with high-dimensional particle filtering, see [14, 25, 115] for example, this work has yet to impact practical data assimilation in, for example, operational weather forecasting. For this reason the *ad hoc* filters, such as 3DVAR, extended Kalman Filter and ensemble Kalman Filter, described in Chapter 4, are of great practical importance. Their analysis is hence an important challenge for applied mathematicians.

- Section 2.6 Data assimilation may be viewed as an inverse problem to determine the signal from the observations. Inverse problems in differential equations are often ill-posed when viewed from a classical non-probabilistic perspective. One reason for this is that the data may not be informative about the whole signal so that many solutions are possible. However taking the Bayesian viewpoint, in which the many solutions are all given a probability, allows for well-posedness to be established. This idea is used for data assimilation problems arising in fluid mechanics in [28], for inverse problems arising in subsurface geophysics in [36, 34] and described more generally in [106]. Well-posedness with respect to changes in the data is of importance in its own right, but also more generally because it underpins other stability results which can be used to control perturbations. In particular the effect of numerical approximation on integration of the forward model can be understood in terms of its effect on the posterior distribution; see [29]. A useful overview of probability metrics, including Hellinger and total variation metrics, is contained in [47].

- Section 2.7. The subject of posterior consistency is central to the theory of statistics in general [113], and within Bayesian statistics in particular [11, 13, 46]. Assessing the quality of data assimilation algorithms is typically performed in the "signal estimation" framework using **identical twin experiments** in which the data is generated from the same model used to estimate the signal; see [61] and the references therein. The idea of assessing "Bayesian quality" has only recently been used within the data assimilation literature; see [71] where this approach is taken for the Navier-Stokes inverse problem formulated in [28]. The evaluation of algorithms by means of forecast skill is enormously influential in the field of numerical weather prediction and drives a great deal of algorithmic selection. The use of information theory to understand the effects of model error , and to evaluate filter performance, is introduced in [81] and [19] respectively. There are also a number of useful **consistency checks** which can be applied to evaluate the computational model and its fit to the data [40, 2, 4]. We discuss the idea of the variant known as *rank histograms* at the end of Chapter 4. If the empirical statistics of the innovations are inconsistent with the assumed model, then they can be used to improve the model used in the future; this is known as *reanalysis.*

2.10 Exercises

1. Consider the map given in Example 2.3 and related program `p1.m`. By experimenting with the code determine approximately the value of α, denoted by α_1, at which the noise-free dynamics changes from convergence to an equilibrium point to convergence to a period 2 solution. Can you find a value of $\alpha = \alpha_2$ for which you obtain a period 4 solution? Can you find a value of $\alpha = \alpha_3$ for which you obtain a non-repeating (chaotic) solution? For the values $\alpha = 2, \alpha_2$ and α_3 compare the trajectories of the dynamical system obtained with the initial condition 1 and with the initial condition 1.1. Comment on what you find. Now fix the initial condition at 1 and consider the same values of α, with and without noise ($\sigma \in \{0, 1\}$). Comment on what you find. Illustrate your answers with graphics. To get interesting displays you will find it helpful to change the value of J (number of iterations) depending upon what you are illustrating.

2. Consider the map given in Example 2.4 and verify the explicit solutions given for $r = 2$ and $r = 4$ in formulae (2.10)–(2.12).

3. Consider the Lorenz '63 model given in Example 2.6. Determine values of $\{\alpha, \beta\}$ for

which (2.16) holds.

4. Consider the Lorenz '96 model given in Example 2.7. Program `p19.m` plots solutions of the model, as well as studying sensitivity to initial conditions. Study the behaviour of the equation for $J = 40, F = 2$, for $J = 40, F = 4$ and report your results. Fix F at 8 and play with the value of the dimension of the system, J. Report your results. Again, illustrate your answers with graphics.

5. Consider the posterior smoothing distribution from Theorem 2.8. Assume that the stochastic dynamics model (2.1) is scalar and defined by $\Psi(v) = av$ for some $a \in \mathbb{R}$ and $\Sigma = \sigma^2$; and that the observation model (2.2) is defined by $h(v) = v$ and $\Gamma = \gamma^2$. Find explicit formulae for $\mathsf{J}(v)$ and $\Phi(v; y)$, assuming that $v_0 \sim N(m_0, \sigma_0^2)$.

6. Consider the posterior smoothing distribution from Theorem 2.11. Assume that the dynamics model (2.3a) is scalar and defined by $\Psi(v) = av$ for some $a \in \mathbb{R}$; and that the observation model (2.2) is defined by $h(v) = v$ and $\Gamma = \gamma^2$. Find explicit formulae for $\mathsf{J}_{\rm det}(v_0)$ and $\Phi_{\rm det}(v_0; y)$, assuming that $v_0 \sim N(m_0, \sigma_0^2)$.

7. Consider the definition of total variation distance given in Definition 1.27. State and prove a theorem analogous to Theorem 2.15, but employing the total variation distance instead of the Hellinger distance.

8. Consider the filtering distribution from section 2.4 in the case where the stochastic dynamics model (2.1) is scalar and defined by $\Psi(v) = av$ for some $a \in \mathbb{R}$ and $\Sigma = \sigma^2$; and that the observation model (2.2) is defined by $h(v) = v$ and $\Gamma = \gamma^2$; and $v_0 \sim N(m_0, \sigma_0^2)$. Demonstrate that the prediction and analysis steps preserve Gaussianity so that $\mu_j = N(m_j, \sigma_j^2)$. Find iterative formulae which update (m_j, σ_j^2) to give $(m_{j+1}, \sigma_{j+1}^2)$.

9. Prove Theorem 2.18.

Chapter 3

Discrete Time: Smoothing Algorithms

The formulation of the data assimilation problem described in the previous chapter is probabilistic, and its computational resolution requires the probing of a posterior probability distribution on signal given data. This probability distribution is on the signal sequence $v = \{v_j\}_{j=0}^J$ when the underlying dynamics is stochastic and given by (2.1); the posterior is specified in Theorem 2.8 and is proportional to $\exp\bigl(-\mathsf{I}(v;y)\bigr)$ given by (2.21). On the other hand, if the underlying dynamics is deterministic and given by (2.3), then the probability distribution is on the initial condition v_0 only; it is given in Theorem 2.11 and is proportional to $\exp\bigl(-\mathsf{I}_{\rm det}(v_0;y)\bigr)$, with $\mathsf{I}_{\rm det}$ given by (2.29). Generically, in this chapter, we refer to the unknown variable as u, and then use v in the specific case of stochastic dynamics, and v_0 in the specific case of deterministic dynamics . The aim of this chapter is to understand $\mathbb{P}(u|y)$. In this regard we will do three things:

- find explicit expressions for the pdf $\mathbb{P}(u|y)$ in the linear, Gaussian setting;
- generate samples $\{u^{(n)}\}_{n=1}^N$ from $\mathbb{P}(u|y)$ by algorithms applicable in the non-Gaussian setting;
- find points where $\mathbb{P}(u|y)$ is maximized with respect to u, for given data y.

In general the probability distributions of interest cannot be described by a finite set of parameters, except in a few simple situations such as the Gaussian scenario where the mean and covariance determine the distribution in its entirety – the **Kalman Smoother**. When the probability distributions cannot be described by a finite set of parameters, an expedient computational approach to approximately representing the measure is through the idea of **Monte Carlo sampling**. The basic idea is to approximate a measure ν by a set of N samples $\{u^{(n)}\}_{n\in\mathbb{Z}^+}$ drawn, or approximately drawn, from ν to obtain the measure $\nu^N \approx \nu$ given by:

$$\nu^N = \frac{1}{N}\sum_{n=1}^N \delta_{u^{(n)}}. \tag{3.1}$$

We may view this as defining a (random) map S^N on measures which takes ν into ν^N. If the $u^{(n)}$ are exact draws from ν then the resulting approximation ν^N converges to the true measure ν as $N \to \infty$. [1] For example if $v = \{v_j\}_{j=0}^J$ is governed by the probability distribution μ_0 defined by the unconditioned dynamics (2.1), and with pdf determined by (2.19), then exact independent samples $v^{(n)} = \{v_j^{(n)}\}_{j=0}^J$ are easy to generate, simply by running the dynamics

[1] Indeed we prove such a result in Lemma 4.7 in the context of the particle filter.

model forward in discrete time. However for the complex probability measures of interest here, where the signal is conditioned on data, exact samples are typically not possible and so instead we use the idea of **Monte Carlo Markov Chain (MCMC)** methods which provide a methodology for generating approximate samples. These methods do not require knowledge of the normalization constant for the measure $\mathbb{P}(u|y)$; as we have discussed, Bayes' formula (1.7) readily delivers $\mathbb{P}(u|y)$ upto normalization, but the normalization itself can be difficult to compute. It is also of interest to simply maximize the posterior probability distribution, to find a single point estimate of the solution, leading to **variational methods**, which we also consider.

Section 3.1 gives explicit formulae for the solution of the smoothing problem in the setting where the stochastic dynamics model is linear, and subject to Gaussian noise, for which the observation operator is linear, and for which the distributions of the initial condition and the observational noise are Gaussian; this is the Kalman smoother. These explicit formulae help to build intuition about the nature of the smoothing distribution. In section 3.2 we provide some background concerning MCMC methods, and in particular, the **Metropolis-Hastings** variant of MCMC, and show how they can be used to explore the posterior distribution. It can be very difficult to sample the probability distributions of interest with high accuracy, because of the two problems of high dimension and sensitive dependence on initial conditions. Whilst we do not claim to introduce the optimal algorithms to deal with these issues, we do discuss such issues in relation to the samplers we introduce, and we provide references to the active research ongoing in this area. Furthermore, although sampling of the posterior distribution may be computationally infeasible in many situations, where possible, it provides an important benchmark solution, enabling other algorithms to be compared against a "gold standard" Bayesian solution.

However, because sampling the posterior distribution can be prohibitively expensive, a widely used computational methodology is simply to find the point which maximizes the probability, using techniques from optimization. These are the variational methods, also known as **4DVAR** and **weak constraint 4DVAR** . We introduce this approach to the problem in section 3.3. In section 3.4 we provide numerical illustrations which showcase the MCMC and variational methods. The chapter concludes in sections 3.5 and 3.6 with bibliographic notes and exercises.

3.1 Linear Gaussian Problems: The Kalman Smoother

The Kalman smoother plays an important role because it is one of the few examples for which the smoothing distribution can be explicitly determined. This explicit characterization occurs because the signal dynamics and observation operator are assumed to be linear. When combined with the Gaussian assumptions on the initial condition for the signal, and on the signal and observational noise, this gives rise to a posterior smoothing distribution which is also Gaussian.

To find formulae for this Gaussian Kalman smoothing distribution we set

$$\Psi(v) = Mv, \quad h(v) = Hv \tag{3.2}$$

for matrices $M \in \mathbb{R}^{n\times n}$, $H \in \mathbb{R}^{m\times n}$ and consider the signal/observation model (2.1), (2.2). Given data $y = \{y_j\}_{j\in\mathbb{J}}$ and signal $v = \{v_j\}_{j\in\mathbb{J}_0}$ we are interested in the probability distribution of $v|y$, as characterized in subsection 2.3.2. By specifying the linear model (3.2), and applying Theorem 2.8 we find the following:

Theorem 3.1 *The posterior smoothing distribution on $v|y$ for the linear stochastic dynamics model (2.1), (2.2), (3.2) with C_0, Σ and Γ symmetric positive-definite is a Gaussian probability measure $\mu = N(m, C)$ on $\mathbb{R}^{|\mathbb{J}_0|\times n}$. The covariance C is the inverse of a symmetric positive-definite block tridiagonal precision matrix*

$$L = \left(\begin{array}{cccccc} L_{11} & L_{12} & & & & \\ L_{21} & L_{22} & L_{23} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & L_{JJ+1} & \\ & & & L_{J+1J} & L_{J+1J+1} & \end{array}\right)$$

with $L_{ij} \in \mathbb{R}^{n\times n}$ given by $L_{11} = C_0^{-1} + M^T\Sigma^{-1}M$, $L_{jj} = H^T\Gamma^{-1}H + M^T\Sigma^{-1}M + \Sigma^{-1}$ for $j = 2, \dots, J$, $L_{J+1,J+1} = H^T\Gamma^{-1}H + \Sigma^{-1}$, $L_{jj+1} = -M^T\Sigma^{-1}$ and $L_{j+1j} = -\Sigma^{-1}M$ for $j = 1, \dots, J$. Furthermore the mean m solves the equation

$$Lm = r,$$

where

$$r_1 = C_0^{-1}m_0, \qquad r_j = H^T\Gamma^{-1}y_{j-1}, \quad j = 2, \cdots, J+1.$$

This mean is also the unique minimizer of the functional

$$\begin{aligned} \mathsf{I}(v;y) &= \frac{1}{2}\bigl|C_0^{-1/2}(v_0 - m_0)\bigr|^2 + \sum_{j=0}^{J-1}\frac{1}{2}\bigl|\Sigma^{-1/2}(v_{j+1} - Mv_j)\bigr|^2 + \sum_{j=0}^{J-1}\frac{1}{2}\bigl|\Gamma^{-1/2}(y_{j+1} - Hv_{j+1})\bigr|^2 \\ &= \frac{1}{2}\bigl|v_0 - m_0\bigr|_{C_0}^2 + \sum_{j=0}^{J-1}\frac{1}{2}|v_{j+1} - Mv_j|_\Sigma^2 + \sum_{j=0}^{J-1}\frac{1}{2}|y_{j+1} - Hv_{j+1}|_\Gamma^2 \end{aligned} \tag{3.3}$$

with respect to v and, as such, is a maximizer, with respect to v, for the posterior pdf $\mathbb{P}(v|y)$.

Proof The proof is based around Lemma 1.6, and identification of the mean and covariance by study of an appropriate quadratic form. From Theorem 2.8 we know that the desired distribution has pdf proportional to $\exp\bigl(-\mathsf{I}(v;y)\bigr)$, where $\mathsf{I}(v;y)$ is given in (3.3). This is a quadratic form in v and we deduce that the inverse covariance L is given by $\partial_v^2\mathsf{I}(v;y)$, the Hessian of I with respect to v. To determine L we note the following identities:

$$\begin{aligned} D_{v_0}^2 I(v;y) &= C_0^{-1} + M^T\Sigma^{-1}M, \\ D_{v_j}^2 I(v;y) &= \Sigma^{-1} + M^T\Sigma^{-1}M + H^T\Gamma^{-1}H, \quad j = 1, \dots, J-1 \\ D_{v_J}^2 I(v;y) &= \Sigma^{-1} + H^T\Gamma^{-1}H, \\ D_{v_j,v_{j+1}}^2 I(v;y) &= -M^T\Sigma^{-1}, \\ D_{v_{j+1},v_j}^2 I(v;y) &= -\Sigma^{-1}M. \end{aligned}$$

We may then complete the square and write

$$\mathsf{I}(v;y) = \frac{1}{2}\langle(v-m), L(v-m)\rangle + q,$$

where q is independent of v. From this it follows that the mean does indeed minimize $\mathsf{I}(v;y)$ with respect to v, and hence maximizes $\mathbb{P}(v|y) \propto \exp\bigl(-\mathsf{I}(v;y)\bigr)$ with respect to v. By differentiating with respect to v we obtain

$$Lm = r, \quad r = -\nabla_v\mathsf{I}(v;y)\Bigr|_{v=0}, \tag{3.4}$$

where ∇_v is the gradient of I with respect to v. This characterization of r gives the desired equation for the mean. Finally we show that L, and hence C, is positive definite symmetric. Clearly L is symmetric and hence so is C. It remains to check that L is strictly positive definite. To see this note that if we set $m_0 = 0$ then

$$\frac{1}{2}\langle v, Lv\rangle = \mathsf{I}(v; 0) \geq 0. \tag{3.5}$$

Moreover, $\mathsf{I}(v; 0) = 0$ with $m_0 = 0$ implies, since $C_0 > 0$ and $\Sigma > 0$,

$$\begin{aligned} v_0 &= 0, \\ v_{j+1} &= Mv_j, \quad j = 0, \dots, J-1, \end{aligned}$$

i.e. $v = 0$. Hence we have shown that $\langle v, Lv\rangle = 0$ implies $v = 0$ and the proof is complete. □

We now consider the Kalman smoother in the case of deterministic dynamics. Application of Theorem 2.11 gives the following:

Theorem 3.2 *The posterior smoothing distribution on $v_0|y$ for the deterministic linear dynamics model* (2.3)*,* (2.2)*,* (3.2) *with C_0 and Γ symmetric positive definite is a Gaussian probability measure $\nu = N(m_{\rm det}, C_{\rm det})$ on $\mathbb{R}^n$. The covariance $C_{\rm det}$ is the inverse of the positive-definite symmetric matrix $L_{\rm det}$ given by the expression*

$$L_{\rm det} = C_0^{-1} + \sum_{j=0}^{J-1} (M^T)^{j+1} H^T \Gamma^{-1} H M^{j+1}.$$

The mean $m_{\rm det}$ solves

$$L_{\rm det} m_{\rm det} = C_0^{-1} m_0 + \sum_{j=0}^{J-1} (M^T)^{j+1} H^T \Gamma^{-1} y_{j+1}.$$

This mean is a minimizer of the functional

$$\mathsf{I}_{\rm det}(v_0; y) = \frac{1}{2}\big|v_0 - m_0\big|_{C_0}^2 + \sum_{j=0}^{J-1} \frac{1}{2}\big|y_{j+1} - HM^{j+1} v_0\big|_{\Gamma}^2 \tag{3.7}$$

with respect to v_0 and, as such, is a maximizer, with respect to v_0, of the posterior pdf $\mathbb{P}(v_0|y)$.

Proof By Theorem 2.11 we know that the desired distribution has pdf proportional to $\exp\big(-\mathsf{I}_{\rm det}(v_0; y)\big)$ given by (3.7). The inverse covariance $L_{\rm det}$ can be found as the Hessian of $\mathsf{I}_{\rm det}$, $L_{\rm det} = \partial_v^2 \mathsf{I}_{\rm det}(v_0; y)$, and the mean $m_{\rm det}$ solves

$$L_{\rm det} m_{\rm det} = -\nabla_v \mathsf{I}_{\rm det}(v_0; y)\Big|_{v_0 = 0}. \tag{3.8}$$

As in the proof of the preceding theorem, we have that

$$\mathsf{I}_{\rm det}(v_0; y) = \frac{1}{2}\langle L_{\rm det}(v_0 - m_{\rm det}), (v_0 - m_{\rm det})\rangle + q$$

where q is independent of v_0; this shows that $m_{\rm det}$ minimizes $\mathsf{I}_{\rm det}(\cdot\,; y)$ and maximizes $\mathbb{P}(\cdot|y)$.

We have thus characterized $L_{\rm det}$ and $m_{\rm det}$ and using this characterization gives the desired expressions. It remains to check that $L_{\rm det}$ is positive definite, since it is clearly symmetric by definition. Positive-definiteness follows from the assumed positive-definiteness of C_0 and Γ since, for any nonzero $v_0 \in \mathbb{R}^n$,

$$\langle v_0, L_{\rm det} v_0\rangle \geq \langle v_0 C_0^{-1} v_0\rangle > 0. \tag{3.9}$$

□

3.2 Markov Chain-Monte Carlo Methods

In the case of stochastic dynamics, equation (2.1), the posterior distribution of interest is the measure μ on $\mathbb{R}^{|\mathbb{J}_0|\times n}$, with density $\mathbb{P}(v|y)$ given in Theorem 2.8; in the case of deterministic dynamics, equation (2.3), it is the measure ν on $\mathbb{R}^n$ with density $\mathbb{P}(v_0|y)$ given in Theorem 2.11. In this section we describe the idea of Markov Chain-Monte Carlo (MCMC) methods for exploring such probability distributions.

We will start by describing the general MCMC methodology, after which we discuss the specific Metropolis-Hastings instance of this methodology. This material makes no reference to the specific structure of our sampling problem; it works in the general setting of creating a Markov chain which is invariant for a an arbitrary measure μ on $\mathbb{R}^\ell$ with pdf ρ. We then describe applications of the Metropolis-Hastings family of MCMC methods to the smoothing problems of noise-free dynamics and noisy dynamics respectively. When describing the generic Metropolis-Hastings methodology we will use u (with indices) to denote the state of the Markov chain and w (with indices) the proposed moves. Thus the current state u and proposed state w live in the space where signal sequences v lie, in the case of stochastic dynamics, and in the space where initial conditions v_0 lie, in the case of deterministic dynamics.

3.2.1. The MCMC Methodology

Recall the concept of a Markov chain$\{u^{(n)}\}_{n\in\mathbb{Z}^+}$ introduced in subsection 1.4.1. The idea of MCMC methods is to construct a Markov chain which is invariant with respect to a given measure μ on $\mathbb{R}^\ell$ and, of particular interest to us, a measure μ with positive Lebesgue density ρ on $\mathbb{R}^\ell$. We now use a superscript n to denote the index of the resulting Markov chain, instead of subscript j, to provide a clear distinction between the Markov chain defined by the stochastic (respectively deterministic) dynamics model (2.1) (respectively (2.3)) and the Markov chains that we will use to sample the posterior distribution on the signal v (respectively initial condition v_0) given data y.

We have already seen that Markov chains allow the computation of averages with respect to the invariant measure by computing the running time-average of the chain – see (1.16). More precisely we have the following theorem (for which it is useful to recall the notation for the iterated kernel p^n from the very end of subsection 1.4.1):

Theorem 3.3 *Assume that, if $u^{(0)} \sim \mu$ with Lebesgue density ρ, then $u^{(n)} \sim \mu$ for all $n \in \mathbb{Z}^+$ so that μ is invariant for the Markov chain. If, in addition, the Markov chain is ergodic, then for any bounded continuous $\varphi : \mathbb{R}^\ell \to \mathbb{R}$,*

$$\frac{1}{N}\sum_{n=1}^{N}\varphi(u^{(n)}) \xrightarrow{a.s.} \mathbb{E}^\mu\varphi(u)$$

for μ a.e. initial condition $u^{(0)}$. In particular, if there is probability measure p on $\mathbb{R}^\ell$ and $\varepsilon > 0$ such that, for all $u \in \mathbb{R}^\ell$ and all Borel sets $A \subseteq \mathcal{B}(\mathbb{R}^\ell)$, $p(u,A) \geq \varepsilon\mathsf{p}(A)$ then, for all $u \in \mathbb{R}^\ell$,

$$d_{\mathrm{TV}}\bigl(p^n(u,\cdot),\mu\bigr) \leq 2(1-\varepsilon)^n. \tag{3.10}$$

Furthermore, there is then $K = K(\varphi) > 0$ such that

$$\frac{1}{N}\sum_{n=1}^{N}\varphi(u^{(n)}) = \mathbb{E}^\mu\varphi(u) + K\xi_N N^{-\frac{1}{2}} \tag{3.11}$$

where ξ_N converges weakly to $N(0,1)$ as $N \to \infty$.

Remark 3.4 *This theorem is the backbone behind MCMC. As we will see, there is a large class of methods which ensure invariance of a given measure μ and, furthermore, these methods are often provably ergodic so that the preceding theorem applies. As with all algorithms in computational science, the optimal algorithm is the one the delivers smallest error for given unit computational cost. In this regard there are two observations to make about the preceding theorem.*

- *The constant K measures the size of the variance of the estimator of $\mathbb{E}^{\mu}\varphi(x)$, multiplied by N. It is thus a surrogate for the error incurred in running MCMC over a finite number of steps. The constant K depends on φ itself, but will also reflect general properties of the Markov chain. For a given MCMC method there will often be tunable parameters whose choice will affect the size of K, without affecting the cost per step of the Markov chain. The objective of choosing these parameters is to minimize the constant K, within a given class of methods all of which have the same cost per step. In thinking about how to do this it is important to appreciate that K measures the amount of correlation in the Markov chain; lower correlation leads to decreased constant K. More precisely, K is computed by integrating the autocorrelation of the Markov chain.*
- *A further tension in designing MCMC methods is in the choice of the class of methods themselves. Some Markov chains are expensive to implement, but the convergence in* (3.11) *is rapid (the constant K can be made small by appropriate choice of parameters), whilst other Markov chains are cheaper to implement, but the convergence in* (3.11) *is slower (the constant K is much larger). Some compromise between ease of implementation and rate of convergence needs to be made.*

♠

3.2.2. Metropolis-Hastings Methods

The idea of Metropolis-Hastings methods is to build an MCMC method for measure μ, by adding an accept/reject test on top of a Markov chain which is easy to implement, but which is not invariant with respect to μ; the accept/reject step is designed to enforce invariance with respect to μ. This is done by enforcing *detailed balance*:

$$\rho(u)p(u,w) = \rho(w)p(w,u) \quad \forall u,w \in \mathbb{R}^{\ell} \times \mathbb{R}^{\ell}. \tag{3.12}$$

Note that integrating with respect to u and using the fact that

$$\int_{\mathbb{R}^{\ell}} p(w,u)du = 1$$

we obtain

$$\int_{\mathbb{R}^{\ell}} \rho(u)p(u,w)du = \rho(w)$$

so that (1.18) is satisfied and density ρ is indeed invariant. We now exhibit an algorithm designed to satisfy detailed balance, by correcting a given Markov chain, which is not invariant with respect to μ, by the addition of an accept/reject mechanism.

We are given a probability density function ρ hence satisfying $\rho : \mathbb{R}^{\ell} \to \mathbb{R}^{+}$, with $\int \rho(u)du = 1$. Now consider a Markov transition kernel $q : \mathbb{R}^{\ell} \times \mathbb{R}^{\ell} \to \mathbb{R}^{+}$ with the property that $\int q(u,w)dw = 1$ for every $u \in \mathbb{R}^{\ell}$. Recall the notation, introduced in subsection

1.4.1, that we use function $q(u,w)$ to denote a pdf and, simultaneously, a probability measure $q(u,dw)$. We create a Markov chain $\{u^{(n)}\}_{n\in\mathbb{N}}$ which is invariant for ρ as follows. Define[2]

$$a(u,w) = 1 \wedge \frac{\rho(w)q(w,u)}{\rho(u)q(u,w)}. \tag{3.13}$$

The algorithm is:

1. Set $n = 0$ and choose $u^{(0)} \in \mathbb{R}^\ell$.
2. $n \to n+1$.
3. Draw $w^{(n)} \sim q(u^{(n-1)}, \cdot)$.
4. Set $u^{(n)} = w^{(n)}$ with probability $a(u^{(n-1)}, w^{(n)})$, $u^{(n)} = u^{(n-1)}$ otherwise.
5. Go to step 2.

At each step in the algorithm there are two sources or randomness: that required for drawing $w^{(n)}$ in step 3; and that required for accepting or rejecting $w^{(n)}$ as the next $u^{(n)}$ in step 4. These two sources of randomness are chosen to be independent of one another. Furthermore, all the randomness at discrete algorithmic time n is independent of randomness at preceding discrete algorithmic times, conditional on $u^{(n-1)}$. Thus the whole procedure gives a Markov chain. If $\mathsf{z} = \{\mathsf{z}^{(j)}\}_{j\in\mathbb{N}}$ is an i.i.d. sequence of $U[0,1]$ random variables then we may write the algorithm as follows:

$$\begin{aligned}
&w^{(n)} \sim q(u^{(n-1)}, \cdot)\\
&u^{(n)} = w^{(n)}\mathbb{I}\big(\mathsf{z}^{(n)} \le a(u^{(n-1)}, w^{(n)})\big) + u^{(n-1)}\mathbb{I}\big(\mathsf{z}^{(n)} > a(u^{(n-1)}, w^{(n)})\big).
\end{aligned}$$

Here $\mathbb{I}$ denotes the indicator function of a set. We let $p : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}^+$ denote the transition kernel of the resulting Markov chain,, and we let p^n denote the transition kernel over n steps; recall that hence $p^n(u, A) = \mathbb{P}(u^{(n)} \in A | u^{(0)} = u)$. Similarly as above, for fixed u, $p^n(u, dw)$ denotes a probability measure on $\mathbb{R}^\ell$ with density $p^n(u,w)$. The resulting algorithm is known as a Metropolis-Hastings MCMC algorithm, and satisfies detailed balance with respect to μ.

Remark 3.5 *The following two observations are central to Metropolis-Hastings MCMC methods.*

- *The construction of Metropolis-Hastings MCMC methods is designed to ensure the* detailed balance *condition* (3.12)*. We will use the condition expressed in this form in what follows later. It is also sometimes written in integrated form as the statement*

$$\int_{\mathbb{R}^\ell\times\mathbb{R}^\ell} f(u,w)\rho(u)p(u,w)dudw = \int_{\mathbb{R}^\ell\times\mathbb{R}^\ell} f(u,w)\rho(w)p(w,u)dudw \tag{3.14}$$

  for all $f : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}$. Once this condition is obtained it follows trivially that the measure μ with density ρ is invariant since, for $f = f(w)$, we obtain

$$\begin{aligned}
\int_{\mathbb{R}^\ell} f(w)\left(\int_{\mathbb{R}^\ell} \rho(u)p(u,w)du\right)dw &= \int_{\mathbb{R}^\ell} f(w)\rho(w)dw \int_{\mathbb{R}^\ell} p(w,u)du\\
&= \int_{\mathbb{R}^\ell} f(w)\rho(w)dw.
\end{aligned}$$

[2] Recall that we use the $\wedge$ operator to denote the minimum between the two real numbers.

Note that $\int_{\mathbb{R}^\ell} \rho(u)p(u,w)du$ is the density of the distribution of the Markov chain after one step, given that it is initially distributed according to density ρ. Thus the preceding identity shows that the expectation of f is unchanged by the Markov chain, if it is initially distributed with density ρ. This means that if the Markov chain is distributed according to measure with density ρ initially then it will be distributed according to the same measure for all algorithmic time.

- *Note that, in order to implement Metropolis-Hastings MCMC methods, it is not necessary to know the normalisation constant for $\rho(\cdot)$ since only its ratio appears in the definition of the acceptance probability a.*

♠

The Metropolis-Hastings algorithm defined above satisfies the following, which requires definition of TV distance given in section 1.3:

Corollary 3.6 *For the Metropolis-Hastings MCMC methods we have that the detailed balance condition* (3.12) *is satisfied and that hence μ is invariant: if $u^{(0)} \sim \mu$ with Lebesgue density ρ, then $u^{(n)} \sim \mu$ for all $n \in \mathbb{Z}^+$. Thus, if the Markov chain is ergodic, then the conclusions of Theorem 3.3 hold.*

We now describe some exemplars of Metropolis-Hastings methods adapted to the data assimilation problem. These are not to be taken as optimal MCMC methods for data assimilation, but rather as examples of how to construct proposal distributions $q(u,\cdot)$ for Metropolis-Hastings methods in the context of data assimilation. In any given application the proposal distribution plays a central role in the efficiency of the MCMC method and tailoring it to the specifics of the problem can have significant impact on efficiency of the MCMC method. Because of the level of generality at which we are presenting the material herein (arbitrary f and h), we cannot discuss such tailoring in any detail.

3.2.3. Deterministic Dynamics

In the case of deterministic dynamics (2.3), the measure of interest is a measure on the initial condition v_0 in $\mathbb{R}^n$. Perhaps the simplest Metropolis-Hastings algorithm is the **Random Walk Metropolis** (RWM) sampler which employs a Gaussian proposal, centred at the current state; we now illustrate this for the case of deterministic dynamics. Recall that the measure of interest is ν with pdf ϱ. Furthermore $\varrho \propto \exp\bigl(-\mathsf{I}_{\rm det}(v_0;y)\bigr)$ as given in Theorem 2.11.

The RWM method proceeds as follows: given that we are at $u^{(n-1)} \in \mathbb{R}^n$, a current approximate sample from the posterior distribution on the initial condition, we propose

$$w^{(n)} = u^{(n-1)} + \beta \iota^{(n-1)} \tag{3.15}$$

where $\iota^{(n-1)} \sim N(0, C_{\rm prop})$ for some symmetric positive-definite proposal covariance $C_{\rm prop}$ and proposal variance scale parameter $\beta > 0$; natural choices for this proposal covariance include the identity I or the prior covariance C_0. Because of the symmetry of such a random walk proposal it follows that $q(w,u) = q(u,w)$ and hence that

$$\begin{aligned} a(u,w) &= 1 \wedge \frac{\varrho(w)}{\varrho(u)} \\ &= 1 \wedge \exp\bigl(\mathsf{I}_{\rm det}(u;y) - \mathsf{I}_{\rm det}(w;y)\bigr). \end{aligned}$$

Remark 3.7 *The expression for the acceptance probability shows that the proposed move to w is accepted with probability one if the value of $\mathsf{I}_{\rm det}(\cdot\,; y)$, the log-posterior, is decreased by moving to w from the current state u. On the other hand, if $\mathsf{I}_{\rm det}(\cdot\,; y)$ increases then the proposed state is accepted only with some probability less than one. Recall that $\mathsf{I}_{\rm det}(\cdot\,; y)$ is the sum of the prior penalization (background) and the model-data misfit functional. The algorithm thus has a very natural interpretation in terms of the data assimilation problem: it biases samples towards decreasing $\mathsf{I}_{\rm det}(\cdot\,; y)$ and hence to improving the fit to both the model and the data in combination.*

The algorithm has two key tuning parameters: the proposal covariance $C_{\rm prop}$ and the scale parameter β. See Remark 3.4, first bullet, for discussion of the role of such parameters. The covariance can encode any knowledge, or guesses, about the relative strength of correlations in the model; given this, the parameter β should be tuned to give an acceptance probability that is neither close to 0 *nor to* 1*. This is because if the acceptance probability is small then successive states of the Markov chain are highly correlated, leading to a large constant K in* (3.11)*. On the other hand if the acceptance probability is close to one then this is typically because β is small, also leading to highly correlated steps and hence to a large constant K in* (3.11)*.* ♠

Numerical results illustrating the method are given in section 3.4.

3.2.4. Stochastic Dynamics

We now apply the Metropolis-Hastings methodology to the data assimilation smoothing problem in the case of the stochastic dynamics model (2.1). Thus the probability measure is on an entire signal sequence $\{v_j\}_{j=0}^{J}$ and not just on v_0; hence it lives on $\mathbb{R}^{|\mathbb{J}_0|\times n}$. It is possible to apply the random walk method to this situation, too, but we take the opportunity to introduce several different Metropolis-Hastings methods, in order to highlight the flexibility of the methodology. Furthermore, it is also possible to take the ideas behind the proposals introduced in this section and apply them in the case of deterministic dynamics.

In what follows recall the measures μ_0 and μ defined in section 2.3, with densities ρ_0 and ρ, representing (respectively) the measure on sequences v generated by (2.1) and the resulting measure when the signal is conditioned on the data y from (2.2). We now construct, via the Metropolis-Hastings methodology, two Markov chains $\{u^{(n)}\}_{n\in\mathbb{N}}$ which are invariant with respect to μ. Hence we need only specify the transition kernel $q(u,w)$, and identify the resulting acceptance probability $a(u,w)$. The sequence $\{w^{(n)}\}_{n\in\mathbb{Z}^+}$ will denote the proposals.

Independence Dynamics Sampler Here we choose the proposal $w^{(n)}$, independently of the current state $u^{(n-1)}$, from the prior μ_0 with density ρ_0. Thus we are simply proposing independent draws from the dynamical model (2.1), with no information from the data used in the proposal. Important in what follows is the observation that

$$\frac{\rho(v)}{\rho_0(v)} \propto \exp(-\Phi(v;y)). \tag{3.16}$$

With the given definition of proposal we have that $q(u,w)=\rho_0(w)$ and hence that

$$\begin{aligned} a(u,w) &= 1\wedge\frac{\rho(w)q(w,u)}{\rho(u)q(u,w)}\\ &= 1\wedge\frac{\rho(w)/\rho_0(w)}{\rho(u)/\rho_0(u)}\\ &= 1\wedge\exp(\Phi(u;y)-\Phi(w;y)). \end{aligned}$$

Remark 3.8 *The expression for the acceptance probability shows that the proposed move to w is accepted with probability one if the value of $\Phi(\cdot\,;y)$ is decreased by moving to w from the current state u. On the other hand, if $\Phi(\cdot\,;y)$ increases then the proposed state is accepted only with some probability less than one, with the probability decreasing exponentially fast with respect to the size of the increase. Recall that $\Phi(\cdot\,;y)$ measures the fit of the signal to the data. Because the proposal builds in the underlying signal model the acceptance probability does not depend on $\mathsf{I}(\cdot\,;y)$, the negative log-posterior, but only the part reflecting the data, namely the negative log-likelihood. In contrast the RWM method, explained in the context of deterministic dynamics, does not build the model into its proposal and hence the accept-reject mechanism depends on the entire log-posterior; see Remark 3.7.* ♠

The Independence Dynamics Sampler does not have any tuning parameters and hence can be very inefficient as there are no parameters to modify in order to obtain a reasonable acceptance probability; as we will see in the illustrations section 3.4 below, the method can hence be quite inefficient because of the resulting frequent rejections. We now discuss this point and an approach to resolve it. The rejections are caused by attempts to move far from the current state, and in particular to proposed states which are based on the underlying stochastic dynamics, but not on the observed data. This typically leads to increases in the model-data misfit functional $\Phi(.\,;y)$ once the Markov chain has found a state which fits the data reasonably well. Even if data is not explicitly used in constructing the proposal, this effect can be ameliorated by making local proposals, which do not move far from the current state. These are exemplified in the following MCMC algorithm.

The pCN Method. It is helpful in what follows to recall the measure ϑ_0 with density π_0 found from μ_0 and ρ_0 in the case where $\Psi \equiv 0$ and given by equation (2.24). We denote the mean by m and covariance by C, noting that $m = (m_0^T, 0^T, \cdots, 0^T)^T$ and that C is block diagonal with first block C_0 and the remainder all being Σ. Thus $\vartheta_0 = N(m, C)$. The basic idea of this method is to make proposals with the property that, if $\Psi \equiv 0$ so that the dynamics is Gaussian and with no time correlation, and if $h \equiv 0$ so that the data is totally uninformative, then the proposal would be accepted with probability one. Making small incremental proposals of this type then leads to a Markov chain which incorporates the effects of $\Psi \neq 0$ and $h \neq 0$ through the accept-reject mechanism. We describe the details of how this works.

Recall the prior on the stochastic dynamics model with density $\rho_0(v) \propto \exp\bigl(-\mathsf{J}(v)\bigr)$ given by (2.19). It will be useful to rewrite π_0 as follows:

$$\pi_0(v) \propto \exp(-\mathsf{J}(v) + F(v)),$$

where

$$F(v) = \sum_{j=0}^{J-1} \left(\frac{1}{2} \left| \Sigma^{-\frac{1}{2}} \Psi(v_j) \right|^2 - \left\langle \Sigma^{-\frac{1}{2}} v_{j+1}, \Sigma^{-\frac{1}{2}} \Psi(v_j) \right\rangle \right). \tag{3.17}$$

We note that

$$\frac{\rho_0(v)}{\pi_0(v)} \propto \exp(-F(v))$$

and hence that, using (2.22),

$$\frac{\rho(v)}{\pi_0(v)} \propto \exp(-\Phi(v;y) - F(v)). \tag{3.18}$$

Recall the Gaussian measure $\vartheta_0 = N(m, C)$ defined via its pdf in (2.24). The pCN method is a variant of random walk type methods, based on the following Gaussian proposal

$$w^{(n)} = m + \bigl(1 - \beta^2\bigr)^{\frac{1}{2}} \Bigl(u^{(n-1)} - m\Bigr) + \beta \iota^{(n-1)}, \tag{3.19}$$

$$\beta \in (0,1], \quad \iota^{(n-1)} \sim N(0,C).$$

Here $\iota^{(n-1)}$ is assumed to be independent of $u^{(n-1)}$.

Lemma 3.9 *Consider the Markov chain*

$$u^{(n)} = m + \left(1-\beta^2\right)^{\frac{1}{2}} \left(u^{(n-1)} - m\right) + \beta \iota^{(n-1)}, \tag{3.20}$$

$$\beta \in (0,1], \quad \iota^{(n-1)} \sim N(0,C)$$

with $\iota^{(n-1)}$ independent of $u^{(n-1)}$. The Markov kernel for this chain $q(u,w)$ satisfies detailed balance (3.12) *with respect to the measure ϑ_0 with density π_0:*

$$\frac{\pi_0(w)q(w,u)}{\pi_0(u)q(u,w)} = 1. \tag{3.21}$$

Proof We show that $\pi_0(u)q(u,w)$ is symmetric in (u,w). To demonstrate this it suffices to consider the quadratic form found by taking the negative of the logarithm of this expression. This is given by

$$\frac{1}{2}|u-m|_C^2 + \frac{1}{2\beta^2}|w-m-(1-\beta^2)^{\frac{1}{2}}(u-m)|_C^2.$$

This is the same as

$$\frac{1}{2\beta^2}|u-m|_C^2 + \frac{1}{2\beta^2}|w-m|_C^2 - \frac{(1-\beta^2)^{\frac{1}{2}}}{\beta^2}\langle w-m, u-m\rangle_C$$

which is clearly symmetric in (u,w). The result follows. □

By use of (3.21) and (3.18) we deduce that the acceptance probability for the MCMC method with proposal (3.19) is

$$\begin{aligned} a(u,w) &= 1 \wedge \frac{\rho(w)q(w,u)}{\rho(u)q(u,w)} \\ &= 1 \wedge \frac{\rho(w)/\pi_0(w)}{\rho(u)/\pi_0(u)} \\ &= 1 \wedge \exp(\Phi(u;y) - \Phi(w;y) + F(u) - F(w)). \end{aligned}$$

Recall that the proposal preserves the underlying Gaussian structure of the stochastic dynamics model; the accept-reject mechanism then introduces non-Gaussianity into the stochastic dynamics model, via F, and introduces the effect of the data, via Φ. By choosing β small, so that $w^{(n)}$ is close to $u^{(n-1)}$, we can make $a(v^{(n-1)}, w^{(n)})$ reasonably large and obtain a usable algorithm. This is illustrated in section 3.4.

Recall from subsection 2.3.3 that, if $\Psi \equiv 0$ (as assumed to define the measure ϑ_0), then the noise sequence $\{\xi_{j-1}\}_{j=1}^{\infty}$ is identical with the signal sequence $\{v_j\}_{j=1}^{\infty}$. More generally, even if $\Psi \neq 0$, the noise sequence $\{\xi_j\}_{j=1}^{\infty}$, together with v_0, a vector which we denote in subsection 2.3.3 by ξ, uniquely determines the signal sequence $\{v_j\}_{j=0}^{\infty}$: see Lemma 2.9. This motivates a different formulation of the smoothing problem for stochastic dynamics where one views the noise sequence and initial condition as the unknown, rather than the signal sequence itself. Here we study the implication of this perspective for MCMC methodology, in the context of the pCN Method, leading to our third sampler within this subsection: the pCN Dynamics Sampler. We now describe this algorithm.

The pCN Dynamics Sampler is so-named because the proposal (implicitly, via the mapping G defined in Lemma 2.9) samples from the dynamics as in the Independence Sampler, while the proposal also includes a parameter β allowing small steps to be taken and chosen to ensure good acceptance probability, as in the pCN Method. The posterior measure we wish to sample is given in Theorem 2.10. Note that this theorem implicitly contains the fact that

$$\vartheta(d\xi) \propto \exp\bigl(-\Phi_{\mathrm{r}}(\xi; y)\bigr)\vartheta_0(d\xi).$$

Furthermore $\vartheta_0 = N(m, C)$ where the mean m and covariance C are as described above for the standard pCN method. We use the pCN proposal (3.19):

$$\zeta^{(n)} = m + \left(1 - \beta^2\right)^{\frac{1}{2}} \left(\xi^{(n-1)} - m\right) + \beta\iota^{(n-1)},$$

and the acceptance probability is given by

$$a(\xi, \zeta) = 1 \wedge \exp\left(\Phi_{\mathrm{r}}(\xi; y) - \Phi_{\mathrm{r}}(\zeta; y)\right).$$

When interpreting this formula it is instructive to note that

$$\Phi_{\mathrm{r}}(\xi; y) = \frac{1}{2}\left|y - \mathcal{G}(\xi)\right|^2_{\Gamma_J} = \frac{1}{2}\left|\Gamma_J^{-\frac{1}{2}}\left(y - \mathcal{G}(\xi)\right)\right|^2 = \Phi\bigl(\mathcal{G}(\xi); y\bigr),$$

and that ξ comprises both v_0 and the noise sequence $\{\xi\}_{j=0}^{J-1}$. Thus the method has the same acceptance probability as the Independence Dynamics Sampler, albeit expressed in terms of initial condition and model noise rather than signal, and also possesses a tunable parameter β; it thus has the nice conceptual interpretation of the acceptance probability that is present in the Independence Dynamics Sampler, as well as the advantage of the pCN method that the proposal variance β may be chosen to ensure a reasonable acceptance probability.

3.3 Variational Methods

Sampling the posterior using MCMC methods can be prohibitively expensive. This is because, in general, sampling involves generating many different points in the state space of the Markov chain. It can be of interest to generate a single point, or small number of points, which represent the salient features of the probability distribution, when this is possible. If the probability is peaked at one, or a small number of places, then simply locating these peaks may be sufficient in some applied contexts. This is the basis for variational methods which seek to maximize the posterior probability, thereby locating such peaks. In practice this boils down to minimizing the negative log-posterior.

We start by illustrating the idea in the context of the Gaussian distributions highlighted in section 3.1 concerning the Kalman smoother. In the case of stochastic dynamics, Theorem 3.1 shows that $\mathbb{P}(v|y)$, the pdf of the posterior distribution, has the form

$$P(v|y) \propto \exp\Bigl(-\frac{1}{2}|v - m|^2_L\Bigr).$$

Now consider the problem

$$v^\star = \mathrm{argmax}_{v \in \mathbb{R}^{|\mathbb{J}_0| \times n}} \mathbb{P}(v|y).$$

From the structure of $\mathbb{P}(v|y)$ we see that

$$v^\star = \mathrm{argmin}_{v \in \mathbb{R}^{|\mathbb{J}_0| \times n}} \mathsf{I}(v; y)$$

where

$$\mathsf{I}(v; y) = \frac{1}{2}|v - m|^2_L = \frac{1}{2}|L^{-\frac{1}{2}}(v - m)|^2.$$

Thus $v^\star = m$, the mean of the posterior. Similarly, using Theorem 3.2, we can show that in the case of deterministic dynamics ,

$$v_0^\star = \mathrm{argmax}_{v_0 \in \mathbb{R}^{|\mathbb{J}_0| \times n}} \mathbb{P}(v_0|y),$$

in the case of deterministic dynamics, is given by $v_0^\star = m_{\mathrm{det}}$.

In this section we show how to characterize peaks in the posterior probability, in the general non-Gaussian case, leading to problems in the calculus of variations. The methods are termed **variational methods**. In the atmospheric sciences these variational methods are referred to as **4DVAR**; this nomenclature reflects the fact that they are variational methods which incorporate data over three spatial dimensions and one temporal dimension (thus four dimensions in total), in order to estimate the state. In Bayesian statistics the methods are called **MAP estimators**: maximum *a posteriori* estimators. It is helpful to realize that the MAP estimator is not, in general, equal to the mean of the posterior distribution. However, in the case of Gaussian posteriors, it is equal to the mean. Computation of the mean of a posterior distribution, in general, requires integrating against the posterior distribution. This can be achieved, via sampling for example, but is typically quite expensive, if sampling is expensive. MAP estimators , in contrast, only require solution of an optimization problem. Unlike the previous section on MCMC methods we do not attempt to overview the vast literature on relevant algorithms (here optimization algorithms); instead references are given in the bibliographic notes of section 3.5.

First we consider the case of stochastic dynamics .

Theorem 3.10 *Consider the data assimilation problem for stochastic dynamics :* (2.1), (2.2)*, with* $\Psi \in C^1(\mathbb{R}^n, \mathbb{R}^n)$ *and* $h \in C^1(\mathbb{R}^n, \mathbb{R}^m)$*. Then:*

- *(i) the infimum of* $\mathsf{I}(\cdot\,; y)$ *given in (2.21) is attained at at least one point* $v^\star$ *in* $\mathbb{R}^{|\mathbb{J}_0| \times n}$*. It follows that the density* $\rho(v) = \mathbb{P}(v|y)$ *on* $\mathbb{R}^{|\mathbb{J}_0| \times n}$ *associated with the posterior probability* μ *given by Theorem 2.8 is maximized at* $v^\star$*;*
- *(ii) furthermore, let* $B(u, \delta)$ *denote a ball in* $\mathbb{R}^{|\mathbb{J}_0| \times n}$ *of radius* δ *and centred at* u*. Then*

$$\lim_{\delta \to 0} \frac{\mathbb{P}^\mu\bigl(B(u_1, \delta)\bigr)}{\mathbb{P}^\mu\bigl(B(u_2, \delta)\bigr)} = \exp\bigl(\mathsf{I}(u_2; y) - \mathsf{I}(u_1; y)\bigr) \quad \text{for all } u_1, u_2 \in \mathbb{R}^{|\mathbb{J}_0| \times n}. \tag{3.22}$$

Proof Note that $\mathsf{I}(\cdot\,; y)$ is non-negative and continuous so that the infimum $\bar{\mathsf{I}}$ is finite and non-negative. To show that the infimum of $\mathsf{I}(\cdot\,; y)$ is attained in $\mathbb{R}^{|\mathbb{J}_0| \times n}$ we let $v^{(n)}$ denote a minimizing sequence. Without loss of generality we may assume that, for all $n \in \mathbb{N}$,

$$\mathsf{I}(v^{(n)}; y) \le \bar{\mathsf{I}} + 1.$$

From the structure of $\mathsf{I}(\cdot\,; y)$ it follows that

$$\begin{aligned} v_0 &= m_0 + C_0^{\frac{1}{2}} r_0, \\ v_{j+1} &= \Psi(v_j) + \Sigma^{\frac{1}{2}} r_{j+1}, \quad j \in \mathbb{Z}^+ \end{aligned}$$

where $\frac{1}{2}|r_j|^2 \le \bar{\mathsf{I}} + 1$ for all $j \in \mathbb{Z}^+$. By iterating and using the inequalities on the $|r_j|$ we deduce the existence of $K > 0$ such that $|v^{(n)}| \le K$ for all $n \in \mathbb{N}$. From this bounded sequence we may extract a convergent subsequence, relabelled to $v^{(n)}$ for simplicity, with limit $v^\star$. By construction we have that $v^{(n)} \to v^\star$ and, for any $\epsilon > 0$, there is $N = N(\epsilon)$ such that

$$\bar{\mathsf{I}} \le \mathsf{I}(v^{(n)}; y) \le \bar{\mathsf{I}} + \epsilon, \quad \forall n \ge N.$$

Hence, by continuity of $\mathsf{I}(\cdot\,;y)$, it follows that

$$\bar{\mathsf{I}} \le \mathsf{I}(v^\star;y) \le \bar{\mathsf{I}} + \epsilon.$$

Since $\epsilon > 0$ is arbitrary it follows that $\mathsf{I}(v^\star;y) = \bar{\mathsf{I}}$. Because

$$\begin{aligned}\mu(dv) &= \frac{1}{Z}\exp(-\mathsf{I}(v;y))dv\\ &= \rho(v)dv\end{aligned}$$

it follows that $v^\star$ also maximizes the posterior pdf ρ.

For the final result we first note that, because Ψ and h are continuously differentiable, the function $I(\cdot\,;y)$ is continuously differentiable. Thus we have

$$\begin{aligned}\mathbb{P}^\mu\big(B(u,\delta)\big) &= \frac{1}{Z}\int_{|v-u|<\delta}\exp\big(-\mathsf{I}(v;y)\big)dv\\ &= \frac{1}{Z}\int_{|v-u|<\delta}\Big(\exp(-\mathsf{I}(u;y)) + e(u;v-u)\Big)dv\end{aligned}$$

where

$$e(u;v-u) = \Big\langle -\int_0^1 D_v I\big(u+s(v-u);y\big)ds, v-u\Big\rangle.$$

As a consequence we have, for $K^\pm > 0$,

$$-K^-|\delta| \le e(u;v-u) \le K^+|\delta|$$

for $u = u_1, u_2$ and $|v-u| < \delta$. Using the preceding we find that, for $E := \exp\big(\mathsf{I}(u_2;y) - \mathsf{I}(u_1;y)\big)$

$$\frac{\mathbb{P}^\mu\big(B(u_1,\delta)\big)}{\mathbb{P}^\mu\big(B(u_2,\delta)\big)} \le E\frac{\int_{|v-u_1|<\delta}\exp\big(K^+|\delta|\big)dv}{\int_{|v-u_2|<\delta}\exp\big(-K^-|\delta|\big)dv} = E\frac{\exp\big(K^+|\delta|\big)}{\exp\big(-K^-|\delta|\big)}.$$

Similarly we have that

$$\frac{\mathbb{P}^\mu\big(B(u_1,\delta)\big)}{\mathbb{P}^\mu\big(B(u_2,\delta)\big)} \ge E\frac{\int_{|v-u_1|<\delta}\exp\big(-K^-|\delta|\big)dv}{\int_{|v-u_2|<\delta}\exp\big(K^+|\delta|\big)dv} = E\frac{\exp\big(-K^-|\delta|\big)}{\exp\big(K^+|\delta|\big)}.$$

Taking the limit $\delta \to 0$ gives the desired result. □

Remark 3.11 *The second statement in Theorem 3.10 may appear a little abstract. It is, however, essentially a complicated way of restating the first statement. To see this fix u_2 and note that the right hand side of* (3.22) *is maximized at point u_1 which minimizes $\mathsf{I}(\cdot\,;y)$. Thus, independently of the choice of* any *fixed u_2, the identity* (3.22) *shows that the probability of a small ball of radius δ centred at u_1 is, approximately, maximized by choosing centres at minimizers of $\mathsf{I}(\cdot\,;y)$. Why, then, do we bother with the second statement? We do so because it makes no reference to Lebesgue density. As such it can be generalized to infinite dimensions, as is required in continuous time for example. We include the second statement for precisely this reason. We also remark that our assumption on continuous differentiability of Ψ and h is stronger than what is needed, but makes for the rather explicit bounds used in the preceding proof and is hence pedagogically desirable.* ♠

The preceding theorem leads to a natural algorithm: compute

$$v = \mathrm{argmin}_{u \in \mathbb{R}^{|\mathbb{J}_0| \times n}} \mathsf{I}(u; y).$$

In applications to meteorology this algorithm is known as **weak constraint 4DVAR**, and we denote this as **w4DVAR** in what follows. The word "weak" in this context is used to indicate that the deterministic dynamics model (2.3a) is not imposed as a strong constraint. Instead the objective functional $\mathsf{I}(\cdot\,; y)$ is minimized; this penalizes deviations from exact satisfaction of the deterministic dynamics model, as well as deviations from the data.

The w4DVAR method generalizes the standard **4DVAR** method which may be derived from w4DVAR in the limit $\Sigma \to 0$ so that the prior on the model dynamics (2.1) is deterministic, but with a random initial condition, as in (2.3). In this case the appropriate minimization is of $\mathsf{I}_{\mathrm{det}}(v_0; y)$ given by (2.29). This has the advantage of being a lower dimensional minimization problem than w4DVAR; however it is often a harder minimization problem, especially when the dynamics is chaotic. The basic 4DVAR algorithm is sometimes called **strong constraint 4DVAR** to denote the fact that the dynamics model (2.3a) is imposed as a strong constraint on the minimization of the model-data misfit with respect to the initial condition; we simply refer to the method as 4DVAR. The following theorem may be proved similarly to Theorem 3.10.

Theorem 3.12 *Consider the data assimilation problem for deterministic dynamics:* (2.3),(2.2) *with* $\Psi \in C^1(\mathbb{R}^n, \mathbb{R}^n)$ *and* $h \in C^1(\mathbb{R}^n, \mathbb{R}^m)$*. Then:*

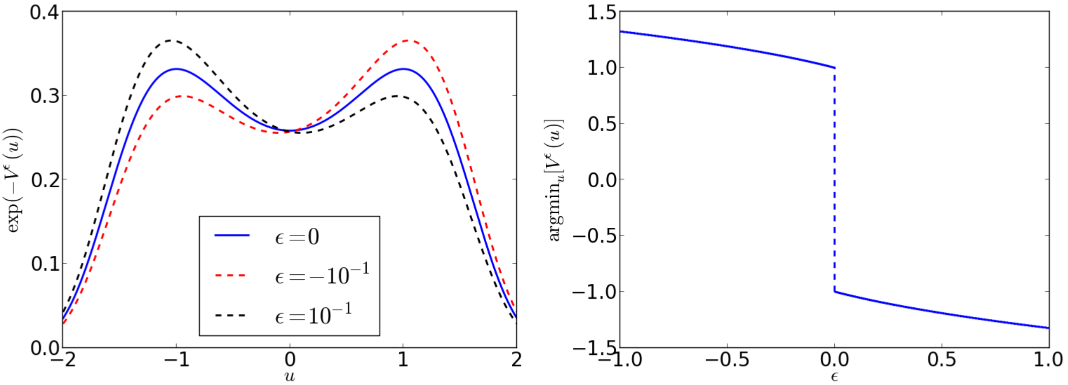
- *(i) the infimum of* $\mathsf{I}_{\mathrm{det}}(\cdot\,; y)$ *given in (2.29) is attained at at least one point* $v_0^\star$ *in* $\mathbb{R}^n$*. It follows that the density* $\varrho(v_0) = \mathbb{P}(v_0|y)$ *on* $\mathbb{R}^n$ *associated with the posterior probability* ν *given by Theorem 2.11 is maximized at* $v_0^\star$*;*
- *(ii) furthermore, if* $B(z, \delta)$ *denotes a ball in* $\mathbb{R}^n$ *of radius* δ*, centred at* z*, then*

$$\lim_{\delta \to 0} \frac{\mathbb{P}^\nu\big(B(z_1, \delta)\big)}{\mathbb{P}^\nu\big(B(z_2, \delta)\big)} = \exp(\mathsf{I}_{\mathrm{det}}(z_2; y) - \mathsf{I}_{\mathrm{det}}(z_1; y)).$$

As in the case of stochastic dynamics we do not discuss optimization methods to perform minimization associated with variational problems; this is because optimization is a well-established and mature research area which is hard to do justice to within the confines of this book. However we conclude this section with an example which illustrates certain advantages of the Bayesian perspective over the optimization or variational perspective. Recall from Theorem 2.15 that the Bayesian posterior distribution is continuous with respect to small changes in the data. In contrast, computation of the global maximizer of the probability may be discontinuous as a function of data. To illustrate this consider the probability measure μ^ϵ on $\mathbb{R}$ with Lebesgue density proportional to $\exp\big(-V^\epsilon(u)\big)$ where

$$V^\epsilon(u) = \frac{1}{4}(1 - u^2)^2 + \epsilon u. \tag{3.23}$$

It is a straightforward application of the methodology behind the proof of Theorem 2.15 to show that μ^ϵ is Lipschitz continuous in ϵ, with respect to the Hellinger metric. Furthermore the methodology behind Theorems 3.10 and 3.12 shows that the probability with respect to this measure is maximized where V^ϵ is minimized. The global minimum, however, changes discontinuously, even though the posterior distribution changes smoothly. This is illustrated in Figure 3.1, where the left hand panel shows the continuous evolution of the probability density function, whilst the right hand-panel shows the discontinuity in the global maximizer of the probability (minimizer of V^ϵ) as ϵ passes through zero. The explanation for this difference between the fully Bayesian approach and MAP estimation is as follows. The measure μ^ϵ has two peaks, for small ϵ, close to ± 1. The Bayesian approach accounts for both of these peaks simultaneously and weights their contribution to expectations. In contrast the MAP estimation approach leads to a global minimum located near $u = -1$ for $\epsilon > 0$ and near $u = +1$ for $\epsilon < 0$, resulting in a discontinuity.



(a) V^ϵ for $\epsilon > 0$, $\epsilon < 0$, and $\epsilon = 0$.

(b) Global minima of V^ϵ as a function of ϵ

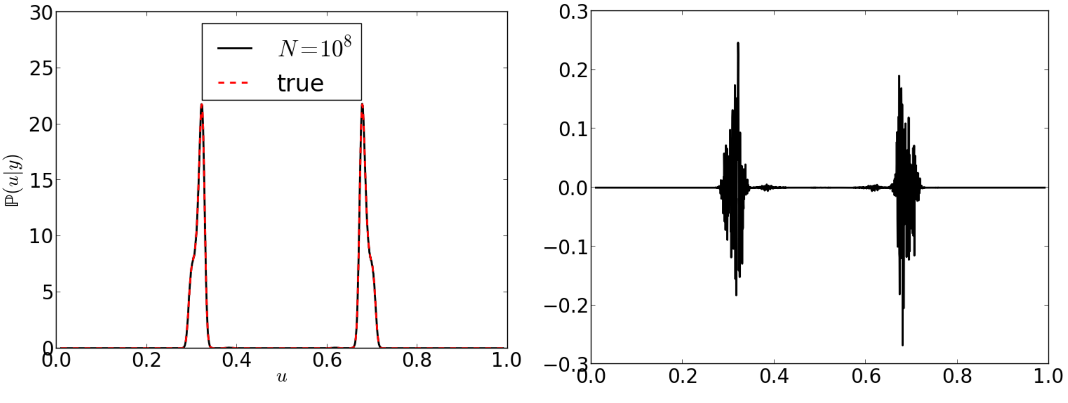
Figure 3.1: Plot of (3.23) shows discontinuity of the global maximum as a function of ϵ.

3.4 Illustrations

We describe a range of numerical experiments which illustrate the application of MCMC methods and variational methods to the smoothing problems which arise in both deterministic and stochastic dynamics .

The first illustration concerns use of the RWM algorithm to study the smoothing distribution for Example 2.4 in the case of deterministic dynamics where our aim is to find $\mathbb{P}(v_0|y)$. Recall Figure 2.13a which shows the true posterior pdf, found by plotting the formula given in Theorem 2.8. We now approximate the true posterior pdf by the MCMC method, using the same parameters, namely $m_0 = 0.5, C_0 = 0.01, \gamma = 0.2$ and $v_0^\dagger = 0.3$. In Figure 3.2 we compare the posterior pdf calculated by the RWM method (denoted by ρ^N, the histogram of the output of the Markov chain) with the true posterior pdf ρ. The two distributions are almost indistinguishable when plotted together in Figure 3.2a; in Figure 3.2b we plot their difference, which as we can see is small, relative to the true value. We deduce that the number of samples used, $N = 10^8$, results here in accurate sampling of the posterior.

We now turn to the use of MCMC methods to sample the smoothing pdf $\mathbb{P}(v|y)$ in the case of stochastic dynamics (2.1), using the Independence Dynamics Sampler and both pCN methods. Before describing application of numerical methods we study the ergodicity of the Independence Dynamics Sampler in a simple, but illustrative, setting. For simplicity assume that the observation operator h is bounded so that, for all $u \in \mathbb{R}^N$, $|h(u)| \le h_{\max}$. Then,



(a) ρ^N vs ρ (b) $\rho^N - \rho$

Figure 3.2: Comparison of the posterior for Example 2.4 for $r = 4$ using random walk metropolis and equation (2.29) directly as in the MATLAB program `p2.m`. We have used $J = 5$ $C_0 = 0.01, m_0 = 0.5$, $\gamma = 0.2$ and true initial condition $v_0 = 0.3$, see also `p3.m` in section 5.2.1. We have used $N = 10^8$ samples from the MCMC algorithm.

recalling the notation $Y_j = \{y_\ell\}_{\ell=1}^{j}$ from the filtering problem, we have

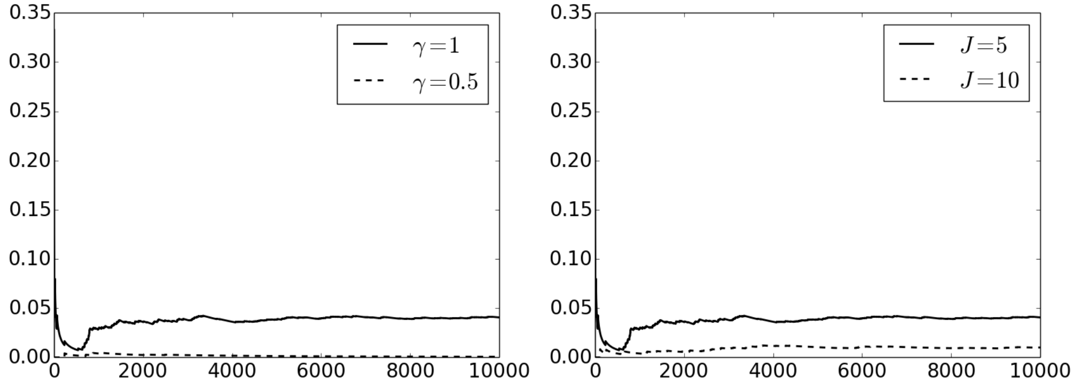
$$\begin{aligned}\Phi(u;y) &\le \sum_{j=0}^{J-1}\big(|\Gamma^{-\frac{1}{2}}y_{j+1}|^2 + |\Gamma^{-\frac{1}{2}}h(u_{j+1})|^2\big)\\ &\le |\Gamma^{-\frac{1}{2}}|^2\Big(\sum_{j=0}^{J-1}|y_{j+1}|^2 + Jh_{\max}^2\Big)\\ &\le |\Gamma^{-\frac{1}{2}}|^2\Big(|Y_J|^2 + Jh_{\max}^2\Big)\\ &=: \Phi_{\max}.\end{aligned}$$

Since $\Phi \ge 0$ this shows that every proposed step is accepted with probability exceeding $e^{-\Phi_{\max}}$ and hence that, since proposals are made with the prior measure μ_0 describing the unobserved stochastic dynamics ,

$$p(u, A) \ge e^{-\Phi_{\max}}\mu_0(A).$$

Thus Theorem 3.3 applies and, in particular, (3.10) and (3.11) hold, with $\varepsilon = e^{-\Phi_{\max}}$, under these assumptions. This positive result about the ergodicity of the MCMC method, also indicates the potential difficulties with the Independence Dynamics Sampler. The Independence Sampler relies on draws from the prior matching the data well. Where the data set is large ($J \gg 1$) or the noise covariance small ($|\Gamma| \ll 1$) this will happen infrequently, because $\Phi_{\max}$ will be large, and the MCMC method will reject frequently and be inefficient. To illustrate this we consider application of the method to the Example 2.3, using the same parameters as in Figure 2.3; specifically we take $\alpha = 2.5$ and $\Sigma = \sigma^2 = 1$. We now sample the posterior distribution and then plot the resulting accept-reject ratio a for the Independence Dynamics Sampler, employing different values of noise Γ and different sizes of the data set J. This is illustrated in Figure 3.3.

In addition, in Figure 3.4, we plot the output, and the running average of the output, projected into the first element of the vector $v^{(k)}$, the initial condition – remember that we



(a) $J = 5$, different values of γ

(b) $\gamma = 1$, different values of J

Figure 3.3: Accept-reject probability of the Independence Sampler for Example 2.3 for $\alpha = 2.5$, $\Sigma = \sigma^2 = 1$ and $\Gamma = \gamma^2$ for different values of γ and J.

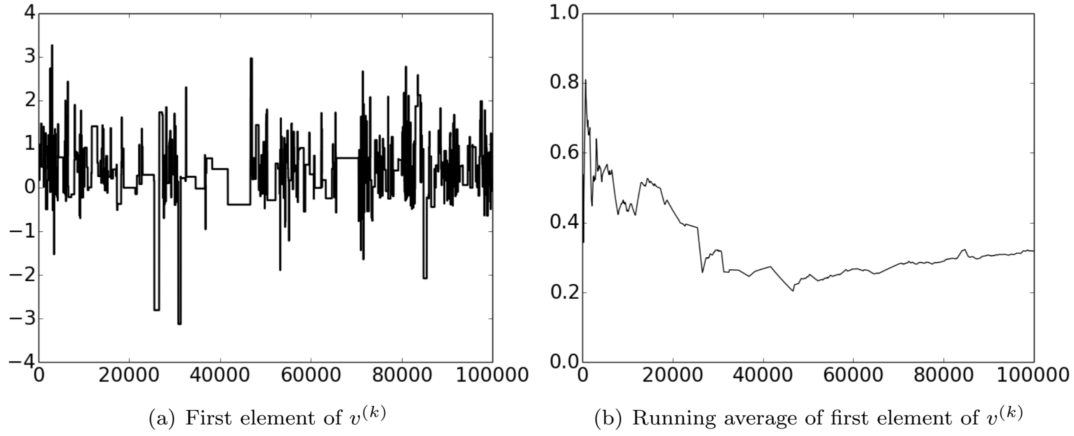
are defining a Markov chain on $\mathbb{R}^{J+1}$ – for $N = 10^5$ steps. Figure 3.4a clearly exhibits the fact that there are many rejections caused by the low average acceptance probability. Figure 3.4b shows that the running average has not converged after 10^5 steps, indicating that the chains needs to be run for longer. If we run the Markov chain over $N = 10^8$ steps then we do get convergence. This is illustrated in Figure 3.5. In Figure 3.5a we see that the running average has converged to its limiting value when this many steps are used. In Figure 3.5b where we plot the marginal probability distribution for the first element of $v^{(k)}$, calculated from this converged Markov chain.

In order to get faster convergence when sampling the posterior distribution we turn to application of the pCN method. Unlike the Independence Dynamics Sampler, this contains a tunable parameter which can vary the size of the proposals. In particular, the possibility of making small moves, with resultant higher acceptance probability, makes this a more flexible method than the Independence Dynamics Sampler. In Figure 3.6 we show application of the pCN sampler, again considering Example 2.3 for $\alpha = 2.5$, $\Sigma = \sigma^2 = 1$ and $\Gamma = \gamma^2 = 1$, with $J = 10$, the same parameters used in Figure 3.4.

In the case that the dynamics are significantly influencing the trajectory, i.e. the regime of large Ψ or small σ, it may be the case that the standard pCN method is not effective, due to large effects of the G term, and the improbability of Gaussian samples being close to samples of the prior on the dynamics. The pCN Dynamics sampler, recall, acts on the space comprising the the initial condition and forcing, both of which are Gaussian under the prior, and so may sometimes have an advantage given that pCN -type methods are based on Gaussian proposals. The use of this method is explored in Figure 3.7 for Example 2.3 for $\alpha = 2.5$, $\Sigma = \sigma^2 = 1$ and $\Gamma = \gamma^2 = 1$, with $J = 10$.

We now turn to variational methods; recall Theorems 3.10 and 3.12 in the stochastic and deterministic cases respectively. In Figure 3.8a we plot the MAP (4DVAR) estimator for our Example 2.1, choosing exactly the same parameters and data as for Figure 2.10a, in the case where $J = 10^2$. In this case the function $\mathsf{I}_{\rm det}(\cdot\,;y)$ is quadratic and has a unique global minimum. A straightforward minimization routine will easily find this: we employed standard MATLAB optimization software initialized at three different points. From all three starting points chosen the algorithm finds the correct global minimizer.

In Figure 3.8b we plot the MAP (4DVAR) estimator for our Example 2.4 for the case



(a) First element of $v^{(k)}$ (b) Running average of first element of $v^{(k)}$

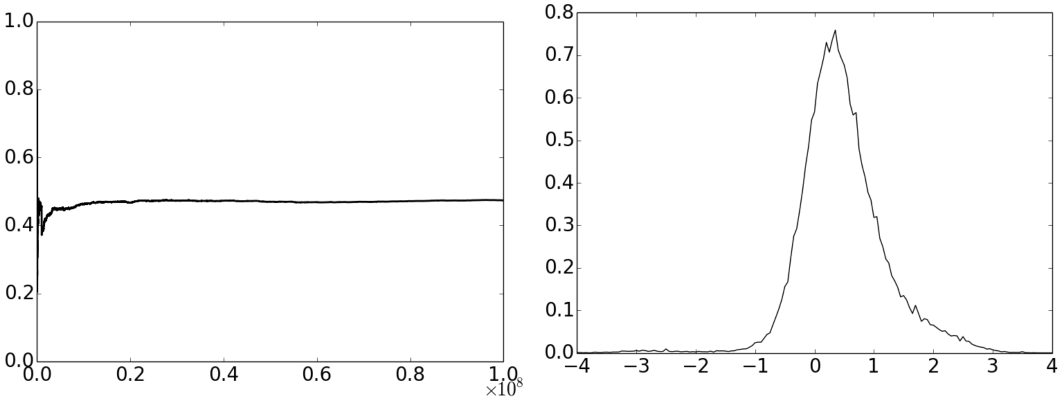
Figure 3.4: Output and running average of the Independence Dynamics Sampler after $K = 10^5$ steps, for Example 2.3 for $\alpha = 2.5$, $\Sigma = \sigma^2 = 1$ and $\Gamma = \gamma^2 = 1$, with $J = 10$, see also `p4.m` in section 5.2.2.

$r = 4$ choosing exactly the same parameters and data as for Figure 2.13. We again employ a MATLAB optimization routine, and we again initialize it at three different points. The value obtained for our MAP estimator depends crucially on the choice of initial condition in our minimization procedure. In particular, on the choices of starting point presented: for the three initializations shown, it is only when we start from 0.2 are we able to find the global minimum of $\mathsf{I}_{\text{det}}(v_0; y)$. By Theorem 3.12 this global minimum corresponds to the maximum of the posterior distribution, and we see that finding the MAP estimator is a difficult task for this problem. Starting with the other two initial conditions displayed we converge to one of the many local minima of $\mathsf{I}_{\text{det}}(v_0; y)$; these local minima are in fact regions of very low probability, as we can see in Figure 2.13a. This illustrates the care required when computing 4DVAR solutions in cases where the forward problem exhibits sensitivity to initial conditions.

Figure 3.9 shows application of the w4DVAR method, or MAP estimator given by Theorem 3.10, in the case of the Example 2.3 with parameters set at $J = 5, \gamma = \sigma = 0.1$. In contrast to the previous example, this is no longer a one-dimensional minimization problem: we are minimizing $\mathsf{I}(v; y)$ given by (2.21) over $v \in \mathbb{R}^6$, given the data $y \in \mathbb{R}^5$. The figure shows that there are at least 2 local minimizers for this problem, with $v^{(1)}$ closer to the truth than $v^{(2)}$, and with $I(v^{(1)}; y)$ considerably smaller that $I(v^{(2)}; y)$. However $v^{(2)}$ has a larger basin of attraction for the optimization software used: many initial conditions lead to $v^{(2)}$, while fewer lead to $v^{(1)}$. Furthermore, whilst we believe that $v^{(1)}$ is the global minimizer, it is difficult to state this with certainty, even for this relatively low-dimensional model. To get greater certainty an exhaustive and expensive search of the six dimensional parameter space would be needed.

3.5 Bibliographic Notes

- The Kalman Smoother from subsection 3.1 leads to a system of linear equations, characterized in Theorem 3.1. These equations are of block tridiagonal form, and may be solved by LU factorization. The Kalman filter corresponds to the LU sweep in this factorization, a fact that was highlighted in [23].



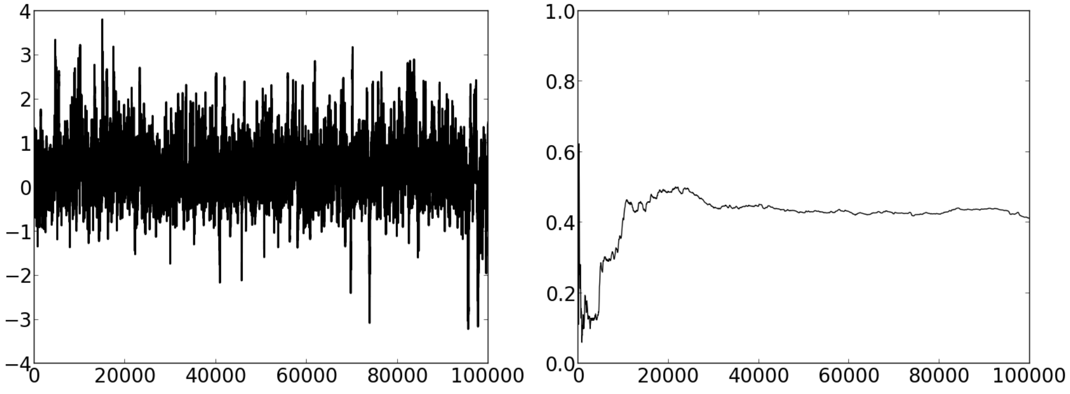
(a) Running average of first element of $v^{(k)}$ (b) Probability density of first element of $v^{(k)}$

Figure 3.5: Running average and probability density of the first element of $v^{(k)}$ for the Independence Dynamics Sampler after $K = 10^8$ steps, for Example 2.3 for $\alpha = 2.5$, $\Sigma = \sigma^2 = 1$ and $\Gamma = \gamma^2$, with $\gamma = 1$ and $J = 10$, see also `p4.m` in section 5.2.2.

- Section 3.2. Monte Carlo Markov Chain methods have a long history, initiated in the 1953 paper [86] and then generalized to an abstract formulation in the 1970 paper [53]. The subject is overviewed from an algorithmic point of view in [74]. Theorem 3.3 is contained in [87], and that reference also contains many other convergence theorems for Markov chains; in particular we note that it is often possible to increase substantially the class of functions φ to which the theorem applies by means of Lyapunov function techniques, which control the tails of the probability distribution. The specific form of the pCN-MCMC method which we introduce here has been chosen to be particularly effective in high dimensions; see [31] for an overview, [16] for the introduction of pCN and other methods for sampling probability measures in infinite dimensions, in the context of conditioned diffusions, and [30] for the application to a data assimilation problem.

  The key point about pCN methods is that the proposal is reversible with respect to an underlying Gaussian measure. Even in the absence of data, if $\Psi \neq 0$ then this Gaussian measure is far from the measure governing the actual dynamics. In contrast, still in the absence of data, this Gaussian measure is *precisely* the measure governing the noise and initial condition, giving the pCN Dynamics Sampler a natural advantage over the standard pCN method. In particular, notice that the acceptance probability is now determined only by the model-data misfit for the pCN Dynamics Sampler, and does *not* have to account for incorporation of the dynamics as it does in the original pCN method; this typically improves the acceptance rate of the pCN Dynamics Sampler over the standard pCN method. Therefore, this method may be preferable, particularly in the case of unstable dynamics. The pCN Dynamics Sampler was introduced in [30] and further trialled in [55]; it shows considerable promise.

  The subject of MCMC methods is an enormous one to which we cannot do justice in this brief presentation. There are two relevant time-scales for the Markov chain: the burn-in time which determines the time to reach part of state-space where most of the probability mass is concentrated, and the mixing time which determines the time taken to fully explore the probability distribution. Our brief overview would not be complete



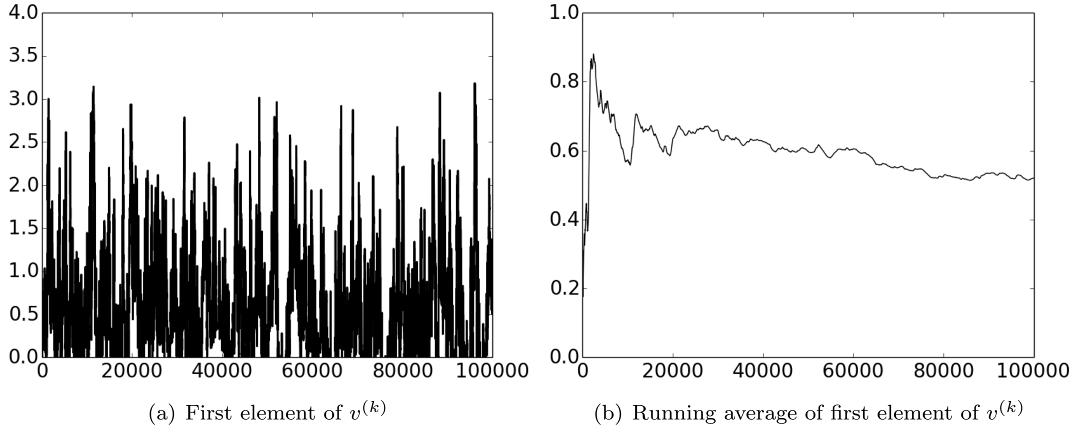
(a) First element of $v^{(k)}$

(b) Running average of first element of $v^{(k)}$

Figure 3.6: Trace-plot and running average of the first element of $v^{(k)}$ for the pCN sampler after $K = 10^5$ steps, for Example 2.3 with $\alpha = 2.5$, $\Sigma = \sigma^2 = 1$ and $\Gamma = \gamma^2 = 1$, with $J = 10$, see also `p5.m` in section 5.2.3.

without a cursory discussion of convergence diagnostics [44] which attempt to ensure that the Markov chain is run long enough to have both burnt-in and mixed. Whilst none of the diagnostics are foolproof, there are many simple tests that can and should be undertaken. The first is to simply study (as we have done in this section) trace plots of quantities of interest (components of the solution, acceptance probabilities) and the running average of these quantities of interest. More sophisticated diagnostics are also available. For example, comparison of the within-chain and between-chain variances of multiple chains beginning from over-dispersed initial conditions is advocated in the works [45, 22]. The authors of those works advise to apply a range of tests based on comparing inferences from individual chains and a mixture of chains. These and other more sophisticated diagnostics are not considered further here, and the reader is referred to the cited works for further details and discussion.

- Section 3.3. Variational Methods, known as 4DVAR in the meteorology community and widely used in practice, have the distinction, when compared with the *ad hoc* non-Gaussian filters described in the next chapter which are also widely used in practice in their EnKF and 3DVAR formulations, of being well-founded statistically: they correspond to the maximum *a posteriori* estimator (MAP estimator) for the fully Bayesian posterior distribution on model state given data [64]. See [122] and the references therein for a discussion of the applied context; see [35] for a more theoretical presentation, including connections to the Onsager-Machlup functional arising in the theory of diffusion processes . The European Centre for Medium-Range Weather Forecasts (ECMWF) runs a weather prediction code based on spectral approximation of continuum versions of Newton's balance laws, together with various sub-grid scale models. Initialization of this prediction code is based on the use of 4DVAR like methods. The conjunction of this computational forward model, together with the use of 4DVAR to incorporate data, results in what is the best weather predictor, worldwide, according to a widely adopted metric by which the prediction skill of forecasts is measured. The subject of algorithms for optimization, which of course underpins variational methods, is vast and we have not attempted to cover it here; we mention briefly that many methods use first derivative information (for example steepest descent methods) and



(a) First element of $v^{(k)}$

(b) Running average of first element of $v^{(k)}$

Figure 3.7: Trace-plot and running average of the first element of $v^{(k)}$ for the pCN dynamics sampler after $K = 10^5$ steps, for Example 2.3 with $\alpha = 2.5$, $\Sigma = \sigma^2 = 1$ and $\Gamma = \gamma^2 = 1$, with $J = 10$, see also p6.m in section 5.2.3.

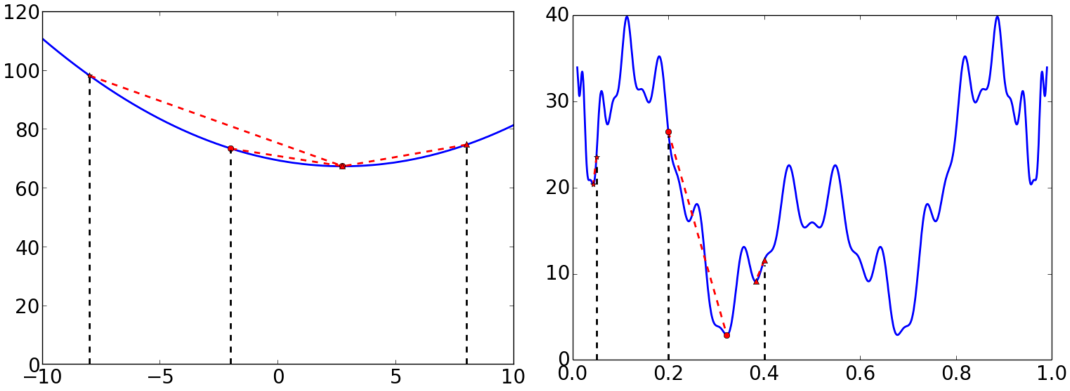
second derivative information (Newton methods); the reader is directed to [91] for details. Derivatives can also be useful in making MCMC proposals, leading the Langevin the hybrid Monte Carlo methods, for example; see [99] and the references therein.

3.6 Exercises

1. Consider the posterior distribution on the initial condition, given by Theorem 2.11, in the case of deterministic dynamics. In the case of Example 2.4, program p2.m plots the prior and posterior distributions for this problem for data generated with true initial condition $v_0 = 0.1$ Why is the posterior distribution concentrating much closer to 0.9 than to the true initial condition at 0.1? Change the mean of the prior from 0.7 to 0.3; what do you observe regarding the effect on the posterior. Explain what you observe. Illustrate your findings with graphics.

2. Consider the posterior distribution on the initial condition, given by Theorem 2.11, in the case of deterministic dynamics. In the case of Example 2.4, program p3.m approximates the posterior distribution for this problem for data generated with true initial condition $v_0 = 0.3$. Why is the posterior distribution in this case approximately symmetric about 0.5? What happens if the mean of the prior is changed from 0.5 to 0.1? Explain what you observe. Illustrate your findings with graphics.

3. Consider the posterior distribution on the initial condition, given by Theorem 2.11, in the case of deterministic dynamics. In the case of Example 2.4, program p3.m approximates the posterior distribution for this problem. Modify the program so that the prior and data are the same as for the first exercise in this section. Compare the approximation to the posterior obtained by use of program p3.m with the true posterior as computed by program p2.m. Carry out similar comparisons for different choices of prior, ensuring that programs p2.m and p3.m share the same prior and the same data. In all cases experiment with the choice of the parameter β in the proposal distribution within p3.m, and determine its effect on the displayed approximation of



(a) $I(v_0; y)$ and its minima for Example 2.1

(b) $I(v_0; y)$ and its minima for Example 2.4

Figure 3.8: Finding local minima of $I(v_0; y)$ for Examples 2.1 and 2.4. The values and the data used are the same as for Figures 2.10a and 2.13b. $(\circ, \star, \Box)$ denote three different initial conditions for the starting the minimization process. $(-8, -2, 8)$ for Example 2.1 and $(0.05, 0.2, 0.4)$ for Example 2.4.

the true posterior computed from `p2.m`. Illustrate your findings with graphics.

4. Consider the posterior distribution on the initial condition, given by Theorem 2.11, in the case of deterministic dynamics. In the case of Example 2.4, program `p3.m` approximates the posterior distribution for this problem. Modify the program so that it applies to Example 2.3. Experiment with the choice of the parameter J, which determines the length of the Markov chain simulation, within `p3.m`. Illustrate your findings with graphics.

5. Consider the posterior distribution on the signal, given by Theorem 2.8, in the case of stochastic dynamics . In the case of Example 2.3, program `p4.m` approximates the posterior distribution for this problem, using the Independence Dynamics Sampler. Run this program for a range of values of γ. Report and explain the effect of γ on the acceptance probability curves.

6. Consider the posterior distribution on the signal, given by Theorem 2.8, in the case of stochastic dynamics . In the case of Example 2.3, program `p5.m` approximates the posterior distribution for this problem, using the pCN sampler. Run this program for a range of values of γ. Report and explain the effect of β on the acceptance probability curves.

7. Consider the posterior distribution on the signal, given by Theorem 2.8, in the case of stochastic dynamics . In the case of Example 2.3, program `p6.m` approximates the posterior distribution for this problem, using the pCN dynamics sampler. Run this program for a range of values of γ. Report and explain the effect of σ and of J on the acceptance probability curves.

8. Consider the MAP estimator for the posterior distribution on the signal, given by Theorem 3.10 in the case of stochastic dynamics . Program `p7.m` finds the MAP estimator for Example 2.3. Increase J to 50 and display your results graphically. Now repeat your experiments for the values $\gamma = 0.01, 0.1$ and 10 and display and discuss your findings.

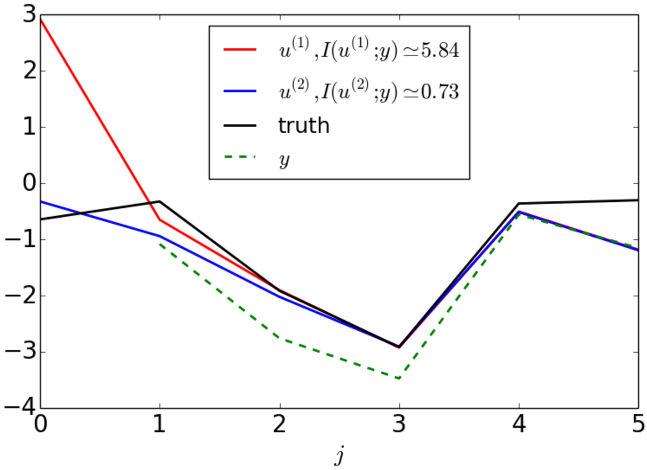


Figure 3.9: Weak constraint 4DVAR for $J = 5, \gamma = \sigma = 0.1$, illustrating two local minimizers $v^{(1)}$ and $v^{(2)}$, see also `p7.m` in section 5.2.5.

Repeat the experiments using the "truth" as the initial condition for the minimization. What effect does this have? Explain this effect.

9. Prove Theorem 3.12.

10. Consider application of the RWM proposal (3.15), applied in the case of stochastic dynamics . Find the form of the Metropolis-Hastings acceptance probability in this case.

11. Consider the family of probability measures μ^ϵ on $\mathbb{R}$ with Lebesgue density proportional to $\exp\bigl(-V^\epsilon(u)\bigr)$ with $V^\epsilon(u)$ given by (3.23). Prove that the family of measure μ^ϵ is locally Lipschitz in the Hellinger metric and in the total variation metric.

Chapter 4

Discrete Time: Filtering Algorithms

In this chapter we describe various algorithms for the filtering problem. Recall from section 2.4 that filtering refers to the sequential update of the probability distribution on the state given the data, as data is acquired, and that $Y_j = \{y_\ell\}_{\ell=1}^{j}$ denotes the data accumulated up to time j. The filtering update from time j to time $j+1$ may be broken into two steps: *prediction* which is based on the equation for the state evolution, using the Markov kernel for the stochastic or deterministic dynamical system which maps $\mathbb{P}(v_j|Y_j)$ into $\mathbb{P}(v_{j+1}|Y_j)$; and *analysis* which incorporates data via Bayes' formula and maps $\mathbb{P}(v_{j+1}|Y_j)$ into $\mathbb{P}(v_{j+1}|Y_{j+1})$. All but one of the algorithms we study (the optimal proposal version of the particle filter) will also reflect these two steps.

We start in section 4.1 with the Kalman filter which provides an exact algorithm to determine the filtering distribution for linear problems with additive Gaussian noise. Since the filtering distribution is Gaussian in this case, the algorithm comprises an iteration which maps the mean and covariance from time j to time $j+1$. In section 4.2 we show how the idea of Kalman filtering may be used to combine dynamical model with data for nonlinear problems; in this case the posterior distribution is not Gaussian, but the algorithms proceed by invoking a Gaussian ansatz in the analysis step of the filter. This results in algorithms which do not provably approximate the true filtering distribution in general; in various forms they are, however, robust to use in high dimension. In section 4.3 we introduce the particle filter methodology which leads to provably accurate estimates of the true filtering distribution but which is, in its current forms, poorly behaved in high dimensions. The algorithms in sections 4.1–4.3 are concerned primarily with stochastic dynamics, but setting $\Sigma = 0$ yields the corresponding algorithms for deterministic dynamics. In section 4.4 we study the long time behaviour of some of the filtering algorithms introduced in the previous sections. Finally, in section 4.5 we present some numerical illustrations and conclude with bibliographic notes and exercises in sections 4.6 and 4.7.

For clarity of exposition we again recall the form of the data assimilation problem. The signal is governed by the model of equations (2.1):

$$\begin{aligned} v_{j+1} &= \Psi(v_j) + \xi_j,\ j \in \mathbb{Z}^+, \\ v_0 &\sim N(m_0, C_0), \end{aligned}$$

where $\xi = \{\xi_j\}_{j\in\mathbb{N}}$ is an i.i.d. sequence, independent of v_0, with $\xi_0 \sim N(0, \Sigma)$. The data is given by equation (2.2):

$$y_{j+1} = h(v_{j+1}) + \eta_{j+1},\ j \in \mathbb{Z}^+,$$

where $h : \mathbb{R}^n \to \mathbb{R}^m$ and $\eta = \{\eta_j\}_{j\in\mathbb{Z}^+}$ is an i.i.d. sequence, independent of (v_0, ξ), with $\eta_1 \sim N(0, \Gamma)$.

4.1 Linear Gaussian Problems: The Kalman Filter

This algorithm provides a sequential method for updating the filtering distribution $\mathbb{P}(v_j|Y_j)$ from time j to time $j+1$, when Ψ and h are linear maps. In this case the filtering distribution is Gaussian and it can be characterized entirely through its mean and covariance. To see this we note that the prediction step preserves Gaussianity by Lemma 1.5; the analysis step preserves Gaussianity because it is an application of Bayes' formula (1.7) and then Lemma 1.6 establishes the required Gaussian property since the log pdf is quadratic in the unknown.

To be concrete we let

$$\Psi(v) = Mv,\ h(v) = Hv \tag{4.2}$$

for matrices $M \in \mathbb{R}^{n\times n}, H \in \mathbb{R}^{m\times n}$. We assume that $m \le n$ and $\mathrm{Rank}(H) = m$. We let (m_j, C_j) denote the mean and covariance of $v_j|Y_j$, noting that this entirely characterizes the random variable since it is Gaussian. We let $(\widehat{m}_{j+1}, \widehat{C}_{j+1})$ denote the mean and covariance of $v_{j+1}|Y_j$, noting that this too completely characterizes the random variable, since it is also Gaussian. We now derive the map $(m_j, C_j) \mapsto (m_{j+1}, C_{j+1})$, using the intermediate variables $(\widehat{m}_{j+1}, \widehat{C}_{j+1})$ so that we may compute the prediction and analysis steps separately. This gives the Kalman filter in a form where the update is expressed in terms of precision rather than covariance.

Theorem 4.1 *Assume that* $C_0, \Gamma, \Sigma > 0$. *Then* $C_j > 0$ *for all* $j \in \mathbb{Z}^+$ *and*

$$\begin{aligned} C_{j+1}^{-1} &= (MC_jM^T + \Sigma)^{-1} + H^T\Gamma^{-1}H, && (4.3a)\\ C_{j+1}^{-1}m_{j+1} &= (MC_jM^T + \Sigma)^{-1}Mm_j + H^T\Gamma^{-1}y_{j+1}. && (4.3b)\end{aligned}$$

Proof We assume for the purposes of induction that $C_j > 0$ noting that this is true for $j = 0$ by assumption. The prediction step is determined by (2.1) in the case $\Psi(\cdot) = M\cdot$:

$$v_{j+1} = Mv_j + \xi_j,\ \xi_j \sim N(0, \Sigma).$$

From this it is clear that

$$\mathbb{E}(v_{j+1}|Y_j) = \mathbb{E}(Mv_j|Y_j) + \mathbb{E}(\xi_j|Y_j).$$

Since ξ_j is independent of Y_j we have

$$\widehat{m}_{j+1} = Mm_j. \tag{4.4}$$

Similarly

$$\begin{aligned}\mathbb{E}\big((v_{j+1} - \widehat{m}_{j+1}) \otimes (v_{j+1} - \widehat{m}_{j+1})|Y_j\big) &= \mathbb{E}\big(M(v_j - m_j) \otimes M(v_j - m_j)|Y_j\big) + \mathbb{E}\big(\xi_j \otimes \xi_j|Y_j\big)\\ &\quad + \mathbb{E}\big(M(v_j - m_j) \otimes \xi_j|Y_j\big) + \mathbb{E}\big(\xi_j \otimes M(v_j - m_j)|Y_j\big).\end{aligned}$$

Again, since ξ_j is independent of Y_j and of v_j, we have

$$\begin{aligned}\widehat{C}_{j+1} &= M\mathbb{E}((v_j - m_j) \otimes (v_j - m_j)|Y_j)M^T + \Sigma\\ &= MC_jM^T + \Sigma. && (4.5)\end{aligned}$$

Note that $\widehat{C}_{j+1} > 0$ because $C_j > 0$ by the inductive hypothesis and $\Sigma > 0$ by assumption.

Now we consider the analysis step . By (2.31), which is just Bayes' formula, and using Gaussianity, we have

$$\exp\Bigl(-\frac{1}{2}\bigl|v-m_{j+1}\bigr|^2_{C_{j+1}}\Bigr) \propto \exp\Bigl(-\frac{1}{2}\bigl|\Gamma^{-\frac{1}{2}}(y_{j+1}-Hv)\bigr|^2-\frac{1}{2}\bigl|\widehat{C}_{j+1}^{-\frac{1}{2}}(v-\widehat{m}_{j+1})\bigr|^2\Bigr) \tag{4.6a}$$

$$= \exp\Bigl(-\frac{1}{2}\bigl|y_{j+1}-Hv\bigr|^2_{\Gamma}-\frac{1}{2}\bigl|v-\widehat{m}_{j+1}\bigr|^2_{\widehat{C}_{j+1}}\Bigr). \tag{4.6b}$$

Equating quadratic terms in v gives, since $\Gamma > 0$ by assumption,

$$C_{j+1}^{-1}=\widehat{C}_{j+1}^{-1}+H^T\Gamma^{-1}H \tag{4.7}$$

and equating linear terms in v gives [1]

$$C_{j+1}^{-1}m_{j+1}=\widehat{C}_{j+1}^{-1}\widehat{m}_{j+1}+H^T\Gamma^{-1}y_{j+1}. \tag{4.8}$$

Substituting the expressions (4.4) and (4.5) for $(\widehat{m}_{j+1},\widehat{C}_{j+1})$ gives the desired result. It remains to verify that $C_{j+1}>0$. From (4.7) it follows, since $\Gamma^{-1}>0$ by assumption and $\widehat{C}_{j+1}>0$ (proved above), that $C_{j+1}^{-1}>0$. Hence $C_{j+1}>0$ and the induction is complete. □

We may now reformulate the Kalman filter using covariances directly, rather than using precisions.

Corollary 4.2 *Under the assumptions of Theorem 4.1, the formulae for the Kalman filter given there may be rewritten as follows:*

$$\begin{aligned}
d_{j+1}&=y_{j+1}-H\widehat{m}_{j+1},\\
S_{j+1}&=H\widehat{C}_{j+1}H^T+\Gamma,\\
K_{j+1}&=\widehat{C}_{j+1}H^TS_{j+1}^{-1},\\
m_{j+1}&=\widehat{m}_{j+1}+K_{j+1}d_{j+1},\\
C_{j+1}&=(I-K_{j+1}H)\widehat{C}_{j+1},
\end{aligned}$$

with $(\widehat{m}_{j+1},\widehat{C}_{j+1})$ given in (4.4), (4.5).

Proof By (4.7) we have

$$C_{j+1}^{-1}=\widehat{C}_{j+1}^{-1}+H^T\Gamma^{-1}H$$

and application of Lemma 4.4 below gives

$$\begin{aligned}
C_{j+1}&=\widehat{C}_{j+1}-\widehat{C}_{j+1}H^T(\Gamma+H\widehat{C}_{j+1}H^T)^{-1}H\widehat{C}_{j+1}\\
&=\Bigl(I-\widehat{C}_{j+1}H^T(\Gamma+H\widehat{C}_{j+1}H^T)^{-1}H\Bigr)\widehat{C}_{j+1}\\
&=(I-\widehat{C}_{j+1}H^TS_{j+1}^{-1}H)\widehat{C}_{j+1}\\
&=(I-K_{j+1}H)\widehat{C}_{j+1}
\end{aligned}$$

as required. Then the identity (4.8) gives

$$\begin{aligned}
m_{j+1}&=C_{j+1}\widehat{C}_{j+1}^{-1}\widehat{m}_{j+1}+C_{j+1}H^T\Gamma^{-1}y_{j+1}\\
&=(I-K_{j+1}H)\widehat{m}_{j+1}+C_{j+1}H^T\Gamma^{-1}y_{j+1}.
\end{aligned} \tag{4.9}$$

[1] We do not need to match the constant terms (with respect to v) since the normalization constant in Bayes theorem deals with matching these.

Now note that, again by (4.7),

$$C_{j+1}(\widehat{C}_{j+1}^{-1} + H^T\Gamma^{-1}H) = I$$

so that

$$\begin{aligned} C_{j+1}H^T\Gamma^{-1}H &= I - C_{j+1}\widehat{C}_{j+1}^{-1} \\ &= I - (I - K_{j+1}H) \\ &= K_{j+1}H. \end{aligned}$$

Since H has rank m we deduce that

$$C_{j+1}H^T\Gamma^{-1} = K_{j+1}.$$

Hence (4.9) gives

$$m_{j+1} = (I - K_{j+1}H)\widehat{m}_{j+1} + K_{j+1}y_{j+1} = \widehat{m}_{j+1} + K_{j+1}d_{j+1}$$

as required. □

Remark 4.3 *The key difference between the Kalman update formulae in Theorem 4.1 and in Corollary 4.2 is that, in the former matrix inversion takes place in the state space, with dimension n, whilst in the latter matrix inversion takes place in the data space, with dimension m. In many applications $m \ll n$, as the observed subspace dimension is much less than the state space dimension, and thus the formulation in Corollary 4.2 is frequently employed in practice. The quantity d_{j+1} is referred to as the* innovation *at time-step $j+1$ and measures the mismatch of the predicted state from the data. The matrix K_{j+1} is known as the* Kalman gain. ♠

The following matrix identity was used to derive the formulation of the Kalman Filter in which inversion takes place in the data space.

Lemma 4.4 Woodbury Matrix Identity *Let $A \in \mathbb{R}^{p\times p}, U \in \mathbb{R}^{p\times q}, C \in \mathbb{R}^{q\times q}$ and $V \in \mathbb{R}^{q\times p}$. If A and C are positive then $A + UCV$ is invertible and*

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\Big(C^{-1} + VA^{-1}U\Big)^{-1}VA^{-1}.$$

4.2 Approximate Gaussian Filters

Here we introduce a family of methods, based on invoking a minimization principle which underlies the Kalman filter, and which has a natural generalization to non-Gaussian problems. The update equation for the Kalman filter mean, (4.8), can be written as

$$m_{j+1} = \arg\min_v \mathsf{I}_{\text{filter}}(v)$$

where

$$\mathsf{I}_{\text{filter}}(v) := \frac{1}{2}|y_{j+1} - Hv|_{\Gamma}^2 + \frac{1}{2}|v - \widehat{m}_{j+1}|_{\widehat{C}_{j+1}}^2; \tag{4.10}$$

here $\widehat{m}_{j+1}$ is calculated from (4.4), and $\widehat{C}_{j+1}$ is given by (4.5). The fact that this minimization principle holds follows from (4.6). (We note that $\mathsf{I}_{\text{filter}}(\cdot)$ in fact depends on j, but we suppress explicit reference to this dependence for notational simplicity.)

Whilst the Kalman filter itself is restricted to linear, Gaussian problems, the formulation via minimization generalizes to nonlinear problems. A natural generalization of (4.10) to the nonlinear case is to define

$$\mathsf{I}_{\text{filter}}(v) := \frac{1}{2}|y_{j+1} - h(v)|_{\Gamma}^2 + \frac{1}{2}|v - \widehat{m}_{j+1}|_{\widehat{C}_{j+1}}^2, \tag{4.11}$$

where

$$\widehat{m}_{j+1} = \Psi(m_j) + \xi_j,$$

and then to set

$$m_{j+1} = \arg\min_{v} \mathsf{I}_{\text{filter}}(v).$$

This provides a family of algorithms for updating the mean, differing depending upon how $\widehat{C}_{j+1}$ is specified. In this section we will consider several choices for this specification, and hence several different algorithms. Notice that the minimization principle is very natural: it enforces a compromise between fitting the model prediction $\widehat{m}_{j+1}$ and the data y_{j+1}.

For simplicity we consider the case where observations are linear and $h(v) = Hv$ leading to the update algorithm $m_j \mapsto m_{j+1}$ defined by

$$\widehat{m}_{j+1} = \Psi(m_j) + \xi_j, \tag{4.12a}$$

$$\mathsf{I}_{\text{filter}}(v) = \frac{1}{2}|y_{j+1} - Hv|_{\Gamma}^2 + \frac{1}{2}|v - \widehat{m}_{j+1})|_{\widehat{C}_{j+1}}^2, \tag{4.12b}$$

$$m_{j+1} = \arg\min_{v} \mathsf{I}_{\text{filter}}(v). \tag{4.12c}$$

This quadratic minimization problem is explicitly solvable and, by the arguments used in deriving Corollary 4.2, we deduce the following update formulae:

$$m_{j+1} = (I - K_{j+1}H)\widehat{m}_{j+1} + K_{j+1}y_{j+1}, \tag{4.13a}$$

$$K_{j+1} = \widehat{C}_{j+1}H^T S_{j+1}^{-1}, \tag{4.13b}$$

$$S_{j+1} = H\widehat{C}_{j+1}H^T + \Gamma. \tag{4.13c}$$

The next three subsections each correspond to algorithms derived in this way, namely by minimizing $\mathsf{I}_{\text{filter}}(v)$, but corresponding to different choices of the model covariance $\widehat{C}_{j+1}$. We also note that in the first two of these subsections we choose $\xi_j \equiv 0$ in equation (4.12a) so that the prediction is made by the noise-free dynamical model; however is not a necessary choice and whilst it is natural for the extended Kalman filter for 3DVAR including random effects in the (4.12a) is also reasonable in some settings. Likewise the ensemble Kalman filter can also be implemented with noise-free prediction models.

We refer to these three algorithms collectively as **approximate Gaussian filters**. This is because they invoke a Gaussian approximation when updating the estimate of the signal via (4.12b). Specifically this update is the correct update for the mean if the assumption that $\mathbb{P}(v_{j+1}|Y_j) = N\big(\widehat{m}_{j+1}), \widehat{C}_{j+1})$ is invoked for the prediction step . In general the approximation implied by this assumption will not be a good one and this can invalidate the statistical accuracy of the resulting algorithms. However the resulting algorithms may still have desirable properties in terms of signal estimation; in subsection 4.4.2 we will demonstrate that this is indeed so.

4.2.1. 3DVAR

This algorithm is derived from (4.13) by simply fixing the model covariance $\widehat{C}_{j+1} \equiv \widehat{C}$ for all j. Thus we obtain

$$\widehat{m}_{j+1} = \Psi(m_j), \tag{4.14a}$$

$$m_{j+1} = (I - KH)\widehat{m}_{j+1} + Ky_{j+1}, \tag{4.14b}$$

$$K = \widehat{C}H^T S^{-1}, \quad S = H\widehat{C}H^T + \Gamma. \tag{4.14c}$$

The nomenclature 3DVAR refers to the fact that the method is variational (it is based on the minimization principle underlying all of the approximate Gaussian methods), and it works sequentially at each fixed time j; as such the minimization, when applied to practical physical problems, is over three spatial dimensions. This should be contrasted with 4DVAR which involves a minimization over all spatial dimensions, as well as time – four dimensions in all.

We now describe two methodologies which generalize 3DVAR by employing model covariances which evolve from step j to step $j+1$: the extended and ensemble Kalman filters. We present both methods in basic form but conclude the section with some discussion of methods widely used in practice to improve their practical performance.

4.2.2. Extended Kalman Filter

The idea of the extended Kalman filter (ExKF) is to propagate covariances according to the linearization of (2.1), and propagate the mean, using (2.3). Thus we obtain, from modification of Corollary 4.2 and (4.4), (4.5)

$$\text{Prediction}\begin{cases} \widehat{m}_{j+1} &= \Psi(m_j), \\ \widehat{C}_{j+1} &= D\Psi(m_j)C_j D\Psi(m_j)^T + \Sigma. \end{cases}$$

$$\text{Analysis}\begin{cases} S_{j+1} &= H\widehat{C}_{j+1}H^T + \Gamma, \\ K_{j+1} &= \widehat{C}_{j+1}H^T S_{j+1}^{-1}, \\ m_{j+1} &= (I - K_{j+1}H)\widehat{m}_{j+1} + K_{j+1}y_{j+1}, \\ C_{j+1} &= (I - K_{j+1}H)\widehat{C}_{j+1}. \end{cases}$$

4.2.3. Ensemble Kalman Filter

The ensemble Kalman filter (EnKF) generalizes the idea of approximate Gaussian filters in a significant way: rather than using the minimization procedure (4.12) to update a single estimate of the *mean*, it is used to generate an *ensemble* of particles which all satisfy the model/data compromise inherent in the minimization; the mean and covariance used in the minimization are then estimated using this ensemble, thereby adding further coupling to the particles, in addition to that introduced by the data.

The EnKF is executed in a variety of ways and we start by describing one of these, the perturbed observation EnKF:

$$\text{Prediction}\begin{cases} \widehat{v}_{j+1}^{(n)} &= \Psi(v_j^{(n)}) + \xi_j^{(n)},\ n = 1, ..., N, \\ \widehat{m}_{j+1} &= \frac{1}{N}\sum_{n=1}^N \widehat{v}_{j+1}^{(n)}, \\ \widehat{C}_{j+1} &= \frac{1}{N-1}\sum_{n=1}^N (\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1})(\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1})^T. \end{cases}$$

$$\text{Analysis}\quad\begin{cases} S_{j+1} &= H\widehat{C}_{j+1}H^T + \Gamma, \\ K_{j+1} &= \widehat{C}_{j+1}H^T S_{j+1}^{-1}, \\ v_{j+1}^{(n)} &= (I - K_{j+1}H)\widehat{v}_{j+1}^{(n)} + K_{j+1}y_{j+1}^{(n)},\ n = 1, ..., N, \\ y_{j+1}^{(n)} &= y_{j+1} + \eta_{j+1}^{(n)},\ n = 1, ..., N. \end{cases}$$

Here $\eta_j^{(n)}$ are i.i.d. draws from $N(0,\Gamma)$ and $\xi_j^{(n)}$ are i.i.d. draws from $N(0,\Sigma)$. Perturbed observation refers to the fact that each particle sees an observation perturbed by an independent draw from $N(0,\Gamma)$. This procedure gives the Kalman Filter in the linear case in the limit of infinite ensemble. Even though the algorithm is motivated through our general approximate Gaussian filters framework, notice that the ensemble is not prescribed to be Gaussian. Indeed it evolves under the full nonlinear dynamics in the prediction step. This fact, together with the fact that covariance matrices are not propagated explicitly, other than through the empirical properties of the ensemble, has made the algorithm very appealing to practitioners.

Another way to motivate the preceding algorithm is to introduce the family of cost functions

$$\mathsf{I}_{\text{filter},n}(v) := \frac{1}{2}|y_{j+1}^{(n)} - Hv|_{\Gamma}^2 + \frac{1}{2}|v - \widehat{v}_{j+1}^{(n)}|_{\widehat{C}_{j+1}}^2. \tag{4.15}$$

The analysis step proceeds to determine the ensemble $\{v_{j+1}^{(n)}\}_{n=1}^N$ by minimizing $\mathsf{I}_{\text{filter},n}$ with $n = 1, \cdots, N$. The set $\{\widehat{v}_{j+1}^{(n)}\}_{n=1}^N$ is found from running the prediction step using the fully nonlinear dynamics. These minimization problems are coupled through $\widehat{C}_{j+1}$ which depends on the entire set of $\{\widehat{v}_j^{(n)}\}_{n=1}^N$. The algorithm thus provides update rules of the form

$$\{v_j^{(n)}\}_{n=1}^N \mapsto \{\widehat{v}_{j+1}^{(n)}\}_{n=1}^N, \qquad \{\widehat{v}_{j+1}^{(n)}\}_{n=1}^N \mapsto \{v_{j+1}^{(n)}\}_{n=1}^N, \tag{4.16}$$

defining approximations of the prediction and analysis steps respectively.

It is then natural to think of the algorithm making the approximations

$$\mu_j \approx \mu_j^N = \frac{1}{N}\sum_{n=1}^N \delta_{v_j^{(n)}}, \qquad \widehat{\mu}_{j+1} \approx \mu_j^N = \frac{1}{N}\sum_{n=1}^N \delta_{\widehat{v}_{j+1}^{(n)}}. \tag{4.17}$$

Thus we have a form of Monte Carlo approximation of the distribution of interest. However, except for linear problems, the approximations given do not, in general, converge to the true distributions μ_j and $\widehat{\mu}_j$ as $N \to \infty$.

4.2.4. Square Root Ensemble Kalman Filters

We now describe another popular variant of the EnKF. The idea of this variant is to define the analysis step in such a way that an ensemble of particles is produced whose empirical covariance *exactly* satisfies the Kalman identity

$$C_{j+1} = (I - K_{j+1}H)\widehat{C}_{j+1} \tag{4.18}$$

which relates the covariances in the analysis step to those in the prediction step. This is done by mapping the mean of the predicted ensemble according to the standard Kalman update, and introducing a linear deterministic transformation of the differences between the particle positions and their mean to enforce (4.18). Doing so eliminates a sampling error inherent in the perturbed observation approach. The resulting algorithm has the following form:

$$\text{Prediction}\begin{cases} \widehat{v}_{j+1}^{(n)} &= \Psi(v_j^{(n)}) + \xi_j^{(n)},\ n = 1, ..., N,\\ \widehat{m}_{j+1} &= \frac{1}{N}\sum_{n=1}^N \widehat{v}_{j+1}^{(n)},\\ \widehat{C}_{j+1} &= \frac{1}{N-1}\sum_{n=1}^N (\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1})(\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1})^T, \end{cases}$$

$$\text{Analysis}\begin{cases} S_{j+1} &= H\widehat{C}_{j+1}H^T + \Gamma,\\ K_{j+1} &= \widehat{C}_{j+1}H^T S_{j+1}^{-1},\\ m_{j+1} &= (I - K_{j+1}H)\widehat{m}_{j+1} + K_{j+1}y_{j+1},\\ v_{j+1}^{(n)} &= m_{j+1} + \zeta_{j+1}^{(n)}. \end{cases}$$

Here the $\{\zeta_{j+1}^{(n)}\}_{n=1}^N$ are designed to have sample covariance $C_{j+1} = (I - K_{j+1}H)\widehat{C}_{j+1}$. There are several ways to do this and we now describe one of them, referred to as the ensemble transform Kalman filter (ETKF).

If we define

$$\widehat{X}_{j+1} = \frac{1}{\sqrt{N-1}}\left[\widehat{v}_{j+1}^{(1)} - \widehat{m}_{j+1}, \dots, \widehat{v}_{j+1}^{(N)} - \widehat{m}_{j+1}\right]$$

then $\widehat{C}_{j+1} = \widehat{X}_{j+1}\widehat{X}_{j+1}^T$. We now seek a transformation T_{j+1} so that, if $X_{j+1} = \widehat{X}_{j+1}T_{j+1}^{\frac{1}{2}}$, then

$$C_{j+1} := X_{j+1}X_{j+1}^T = (I - K_{j+1}H)\widehat{C}_{j+1}. \tag{4.19}$$

Note that the X_{j+1} (resp. the $\widehat{X}_{j+1}$) correspond to Cholesky factors of the matrices C_{j+1} (resp. $\widehat{C}_{j+1}$) respectively. We may now define the $\{\zeta_{j+1}^{(n)}\}_{n=1}^N$ by

$$X_{j+1} = \frac{1}{\sqrt{N-1}}\left[\zeta_{j+1}^{(1)}, \dots, \zeta_{j+1}^{(N)}\right].$$

We now demonstrate how to find an appropriate transformation T_{j+1}. We assume that T_{j+1} is symmetric and positive-definite and the standard matrix square-root is employed. Choosing

$$T_{j+1} = \left[I + (H\widehat{X}_{j+1})^T\Gamma^{-1}(H\widehat{X}_{j+1})\right]^{-1}$$

we see that

$$\begin{aligned} X_{j+1}X_{j+1}^T &= \widehat{X}_{j+1}T_{j+1}\widehat{X}_{j+1}^T\\ &= \widehat{X}_{j+1}\left[I + (H\widehat{X}_{j+1})^T\Gamma^{-1}(H\widehat{X}_{j+1})\right]^{-1}\widehat{X}_{j+1}^T\\ &= \widehat{X}_{j+1}\left\{I - (H\widehat{X}_{j+1})^T\left[(H\widehat{X}_{j+1})(H\widehat{X}_{j+1})^T + \Gamma\right]^{-1}(H\widehat{X}_{j+1})\right\}\widehat{X}_{j+1}^T\\ &= (I - K_{j+1}H)\widehat{C}_{j+1} \end{aligned}$$

as required, where the transformation between the second and third lines is justified by Lemma 4.4. It is important to ensure that $\mathbf{1}$, the vector of all ones, is an eigenvector of the transformation T_{j+1}, and hence of $T_{j+1}^{\frac{1}{2}}$, so that the mean of the ensemble is preserved. This is guaranteed by T_{j+1} as defined.

4.3 The Particle Filter

In this section we introduce an important class of filtering methods known as *particle filters*. In contrast to the filters introduced in the preceding section, the particle filter can be *proved* to reproduce the true posterior filtering distribution in the large particle limit and, as such, has a privileged places amongst all the filters introduced in this book. We will describe the method in its basic form – the *bootstrap filter* – and then give a proof of convergence. It is important to appreciate that the form of particle filter introduced here is far from state-of-the-art and that far more sophisticated versions are used in practical applications. Nonetheless, despite this sophistication, particle filters do not perform well in applications such as those arising in geophysical applications of data assimilation, because the data in those applications places very strong constraints on particle locations, making efficient algorithms very hard to design. It is for this reason that we have introduced particle filters after the approximate Gaussian filters introduced in the preceding section. The filters in the preceding section tend to be more robust to data specifications. However they do all rely on the invocation of ad hoc Gaussian assumptions in their derivation and hence do not provably produce the correct posterior filtering distribution, notwithstanding their ability, in partially observed small noise scenarios, to correctly identify the signal itself, as in Theorem 4.10. Because it can provably reproduce the correct filtering distribution, the particle filter thus plays an important role, conceptually, even though it is not, in current form, a practical algorithm in geophysical applications. With further improvements it may, in time, form the basis for practical algorithms in geophysical applications.

4.3.1. The Basic Approximation Scheme

All probability measures which possess density with respect to Lebesgue measure can be approximated by a finite convex combination of Dirac probability measures; an example of this is the **Monte Carlo sampling** idea that we described at the start of Chapter 3, and also underlies the ensemble Kalman filter of subsection 4.2.3. In practice the idea of approximation by a convex combination of probability measures requires the determination of the locations and weights associated with these Dirac measures. Particle filters are sequential algorithms which use this idea to approximate the true filtering distribution $\mathbb{P}(v_j|Y_j)$.

Basic Monte Carlo, as in (3.1), and the ensemble Kalman filter, as in (4.17), correspond to approximation by equal weights. Recall μ_j, the probability measure on $\mathbb{R}^n$ corresponding to the density $\mathbb{P}(v_j|Y_j)$, and $\widehat{\mu}_{j+1}$, the probability measure on $\mathbb{R}^n$ corresponding to the density $\mathbb{P}(v_{j+1}|Y_j)$. The basic form of the particle filter proceeds by allowing the weights to vary and by finding N-particle Dirac measure approximations of the form

$$\mu_j \approx \mu_j^N := \sum_{n=1}^{N} w_j^{(n)} \delta_{v_j^{(n)}}, \qquad \widehat{\mu}_{j+1} \approx \widehat{\mu}_{j+1}^N := \sum_{n=1}^{N} \widehat{w}_{j+1}^{(n)} \delta_{\widehat{v}_{j+1}^{(n)}}. \tag{4.20}$$

The weights must sum to one. The approximate distribution μ_j^N is completely defined by particle positions $v_j^{(n)}$ and weights $w_j^{(n)}$, and the approximate distribution $\widehat{\mu}_{j+1}^N$ is completely defined by particle positions $\widehat{v}_{j+1}^{(n)}$ and weights $\widehat{w}_{j+1}^{(n)}$. Thus the objective of the method is to find update rules

$$\{v_j^{(n)}, w_j^{(n)}\}_{n=1}^N \mapsto \{\widehat{v}_{j+1}^{(n)}, \widehat{w}_{j+1}^{(n)}\}_{n=1}^N, \qquad \{\widehat{v}_{j+1}^{(n)}, \widehat{w}_{j+1}^{(n)}\}_{n=1}^N \mapsto \{v_{j+1}^{(n)}, w_{j+1}^{(n)}\}_{n=1}^N \tag{4.21}$$

defining the prediction and analysis approximations respectively; compare this with (4.16) for the EnKF where the particle weights are uniform and only the positions are updated.

Defining the updates for the particle filter may be achieved by an application of sampling, for the prediction step , and of Bayesian probability, for the analysis step.

Recall the prediction and analysis formulae from (2.30) and (2.31) which can be summarized as

$$\mathbb{P}(v_{j+1}|Y_j) = \int_{\mathbb{R}^n} \mathbb{P}(v_{j+1}|v_j)\mathbb{P}(v_j|Y_j)dv_j, \tag{4.22a}$$

$$\mathbb{P}(v_{j+1}|Y_{j+1}) = \frac{\mathbb{P}(y_{j+1}|v_{j+1})\mathbb{P}(v_{j+1}|Y_j)}{\mathbb{P}(y_{j+1}|Y_j)}. \tag{4.22b}$$

We may rewrite (4.22) as

$$\widehat{\mu}_{j+1}(\cdot) = (P\mu_j)(\cdot) := \int_{\mathbb{R}^n} \mathbb{P}(\cdot|v_j)\mu_j(dv_j) \tag{4.23a}$$

$$\frac{d\mu_{j+1}}{d\widehat{\mu}_{j+1}}(v_{j+1}) = \frac{\mathbb{P}(y_{j+1}|v_{j+1})}{\mathbb{P}(y_{j+1}|Y_j)}. \tag{4.23b}$$

Writing the update formulae this way is important for us because they then make sense in the absence of Lebesgue densities; in particular we can use them in situations where Dirac masses appear, as they do in our approximate probability measures. The formula (4.23b) for the *density* or *Radon-Nikodym derivative* of μ_{j+1} with respect to that of $\widehat{\mu}_{j+1}$ has a straightforward interpretation: the righthand-side quantifies how to reweight expectations under $\widehat{\mu}_{j+1}$ so that they become expectations under μ_{j+1}. To be concrete we may write

$$\mathbb{E}^{\mu_{j+1}}\varphi(v_{j+1}) = \mathbb{E}^{\widehat{\mu}_{j+1}}\Big(\frac{d\mu_{j+1}}{d\widehat{\mu}_{j+1}}(v_{j+1})\varphi(v_{j+1})\Big).$$

4.3.2. Sequential Importance Resampling

The simplest particle filter, which is based on *sequential importance resampling* is now described. We start by assuming that we have an approximation μ_j^N given by (4.20) and explain how to evolve the weights $\{v_j^{(n)}, w_j^{(n)}\}_{n=1}^N$ into $\{v_j^{(n+1)}, w_j^{(n+1)}\}_{n=1}^N$, via $\{\widehat{v}_{j+1}^{(n)}, \widehat{w}_{j+1}^{(n)}\}_{n=1}^N$ as in (4.21).

Prediction In this step we approximate the prediction phase of the Markov chain. To do this we simply draw $\widehat{v}_{j+1}^{(n)}$ from the kernel p of the Markov chain (2.1a) started from $v_j^{(n)}$. Thus the relevant kernel is $p(v_j, v_{v+1}) = \mathbb{P}(v_{j+1}|v_j)$. We then have $\widehat{v}_{j+1}^{(n)} \sim p(v_j^{(n)}, \cdot)$. We leave the weights of the approximation unchanged so that $\widehat{w}_{j+1}^{(n)} = w_j^{(n)}$. From these new particles and (in fact unchanged) weights we have the particle approximation

$$\widehat{\mu}_{j+1}^N = \sum_{n=1}^N w_j^{(n)} \delta_{\widehat{v}_{j+1}^{(n)}}. \tag{4.24}$$

Analysis In this step we approximate the incorporation of data via Bayes' formula. Define $g_j(v)$ by

$$g_j(v_{j+1}) \propto \mathbb{P}(y_{j+1}|v_{j+1}), \tag{4.25}$$

where the constant of proportionality is, for example, the normalization for the Gaussian, and is hence independent of both y_{j+1} and v_{j+1}. We now apply Bayes' formula in the form (4.23b). Thus we obtain

$$\mu_{j+1}^N = \sum_{n=1}^N w_{j+1}^{(n)} \delta_{\widehat{v}_{j+1}^{(n)}} \tag{4.26}$$

where

$$w_{j+1}^{(n)} = \widetilde{w}_{j+1}^{(n)} \Big/ \left(\sum_{n=1}^{N} \widetilde{w}_{j+1}^{(n)} \right), \qquad \widetilde{w}_{j+1}^{(n)} = g_j\big(\widehat{v}_{j+1}^{(n)}\big) w_j^{(n)}. \tag{4.27}$$

The first equation in the preceding is required for normalization. Thus in this step we do not change the particle positions, but we reweight them.

Resampling The algorithm as described is deficient in two regards, both of which can be dealt with by introducing a re-sampling step into the algorithm. Firstly, the initial measure μ_0 for the true filtering distribution will not typically be made up of a combination of Dirac measures. Secondly, the method can perform poorly if one of the particle weights approaches 1 (and then all others approach 0). The effect of the first can be dealt with by sampling the initial measure and approximating it by an equally weighted (by N^{-1}) sum of Dirac measures at the samples. The second can be ameliorated by drawing a set of N particles from the measure (4.26) and assigning weight N^{-1} to each; this has the effect of multiplying particles with high weights and killing particles with low weights.

Putting together the three preceding steps leads to the following algorithm; for notational convenience we use Y_0 to denote the empty vector (no observations at the start):

1. Set $j = 0$ and $\mu_0^N(dv_0) = \mu_0(dv_0)$.
2. Draw $v_j^{(n)} \sim \mu_j^N$, $n = 1, \dots, N$.
3. Set $w_j^{(n)} = 1/N$, $n = 1, \dots, N$; redefine $\mu_j^N := \sum_{n=1}^N w_j^{(n)} \delta_{v_j^{(n)}}$.
4. Draw $\widehat{v}_{j+1}^{(n)} \sim p(v_j^{(n)}|\cdot)$.
5. Define $w_{j+1}^{(n)}$ by (4.27) and $\mu_{j+1}^N := \sum_{n=1}^N w_{j+1}^{(n)} \delta_{\widehat{v}_{j+1}^{(n)}}$.
6. $j + 1 \to j$.
7. Go to step 2.

This algorithm is conceptually intuitive, proposing that each particle moves according to the dynamics of the underlying model itself, and is then re-weighted according to the likelihood of the proposed particle, i.e. according to the data. This sequential importance resampling filter is also sometimes termed the **bootstrap filter**. We will comment on important improvements to this basic algorithm in the the following section and in the bibliographic notes. Here we prove convergence of this basic method, as the number of particles goes to infinity, thereby demonstrating the potential power of the bootstrap filter and more sophisticated variants on it.

Recall that, by (2.32), the true filtering distribution simply satisfies the iteration

$$\mu_{j+1} = L_j P \mu_j, \quad \mu_0 = N(m_0, C_0), \tag{4.28}$$

where P corresponds to moving a point currently at v according to the Markov kernel $p(\cdot|v)$ describing the dynamics given by (2.1a) and L_j denotes the application of Bayes' formula with likelihood proportional to $g_j(\cdot)$ given by (4.25). Recall also the sampling operator S^N defined by (3.1). It is then instructive to write the particle filtering algorithm which approximates (4.28) in the following form:

$$\mu_{j+1}^N = L_j S^N P \mu_j^N, \quad \mu_0^N = \mu_0. \tag{4.29}$$

There is a slight trickery here in writing application of the sampling S^N *after* application of P, but some reflection shows that this is well-justified: applying P followed by S^N can be shown, by first conditioning on the initial point and sampling with respect to P, and then sampling over the distribution of the initial point, to be the algorithm as defined.

Comparison of (4.28) and (4.29) shows that analyzing the particle filter requires estimation of the error induced by application of S^N (the *resampling error*) together with estimation of the rate of accumulation of this error in time under the application of L_j and P. We now build the tools to allow us to do this. The operators L_j, P and S^N map the space $\mathcal{P}(\mathbb{R}^n)$ of probability measures on $\mathbb{R}^n$ into itself according to the following:

$$(L_j\mu)(dv) = \frac{g_j(v)\mu(dv)}{\int_{\mathbb{R}^n} g_j(v)\mu(dv)}, \tag{4.30a}$$

$$(P\mu)(dv) = \int_{\mathbb{R}^n} p(v', dv)\mu(dv'), \tag{4.30b}$$

$$(S^N\mu)(dv) = \frac{1}{N}\sum_{n=1}^{N} \delta_{v^{(n)}}(dv), \quad v^{(n)} \sim \mu \quad \text{i.i.d.}. \tag{4.30c}$$

Notice that both L_j and P are deterministic maps, whilst S^N is random. Let $\mu = \mu_\omega$ denote, for each ω, an element of $\mathcal{P}(\mathbb{R}^n)$. If we then assume that ω is a random variable describing the randomness required to define the sampling operator S^N, and let $\mathbb{E}^\omega$ denote expectation over ω, then we may define a "root mean square" distance $d(\cdot,\cdot)$ between two random probability measures μ_ω, ν_ω, as follows:

$$d(\mu,\nu) = \sup\nolimits_{|f|_\infty \le 1} \sqrt{\mathbb{E}^\omega|\mu(f) - \nu(f)|^2}.$$

Here we have used the convention that $\mu(f) = \int_{\mathbb{R}^n} f(v)\mu(dv)$ for measurable $f : \mathbb{R}^n \to \mathbb{R}$, and similar for ν. Furthermore

$$|f|_\infty = \sup_u |f(u)|.$$

This distance does indeed generate a metric and, in particular, satisfies the triangle inequality. Note also that, in the absence of randomness within the measures, the metric satisfies $d(\mu,\nu) = 2d_{\text{TV}}(\mu,\nu)$, by (1.12); that is, it reduces to the total variation metric . In our context the randomness within the probability measures comes from the sampling operator S^N used to define the numerical approximation.

Theorem 4.5 *We assume in the following that there exists $\kappa \in (0,1]$ such that for all $v \in \mathbb{R}^n$ and $j \in \mathbb{N}$*

$$\kappa \le g_j(v) \le \kappa^{-1}.$$

Then

$$d(\mu_J^N, \mu_J) \le \sum_{j=1}^{J} (2\kappa^{-2})^j \frac{1}{\sqrt{N}}.$$

Proof The desired result is proved below in a straightforward way from the following three facts, whose proof we postpone to three lemmas at the end of the section:

$$\sup_{\mu\in\mathcal{P}(\mathbb{R}^n)} d(S^N\mu,\mu) \le \frac{1}{\sqrt{N}}, \tag{4.31a}$$

$$d(P\nu, P\mu) \le d(\nu,\mu), \tag{4.31b}$$

$$d(L_j\nu, L_j\mu) \le 2\kappa^{-2} d(\nu,\mu). \tag{4.31c}$$

By the triangle inequality we have, for $\nu_j^N = P\mu_j^N$,

$$\begin{aligned} d(\mu_{j+1}^N, \mu_{j+1}) &= d(L_j S^N P\mu_j^N, L_j P\mu_j) \\ &\le d(L_j P\mu_j^N, L_j P\mu_j) + d(L_j S^N P\mu_j^N, L_j P\mu_j^N) \\ &\le 2\kappa^{-2}\Big(d(\mu_j^N, \mu_j) + d(S^N\nu_j^N, \nu_j^N)\Big) \\ &\le 2\kappa^{-2}\Big(d(\mu_j^N, \mu_j) + \frac{1}{\sqrt{N}}\Big). \end{aligned}$$

Iterating, after noting that $\mu_0^N = \mu_0$, gives the desired result. □

Remark 4.6 *This important theorem shows that the particle filter reproduces the true filtering distribution, in the large particle limit. We make some comments about this.*

- *This theorem shows that, at any fixed discrete time j, the filtering distribution μ_j is well-approximated by the bootstrap filtering distribution μ_j^N in the sense that, as the number of particles $N \to \infty$, the approximating measure converges to the true measure. However, since $\kappa < 1$, the number of particles required to decrease the upper bound on the error beneath a specified tolerance grows with J.*

- *If the likelihoods have a small lower bound then the constant in the convergence proof may be prohibitively expensive, requiring an enormous number of particles to obtain a small error. This is similar to the discussion concerning the Independence Dynamics Sampler in section 3.4 where we showed that large values in the potential Φ lead to slow convergence of the Markov chain, and the resultant need for a large number of samples.*

- *In fact in many applications the likelihoods g_j may not be bounded from above or below, uniformly in j, and more refined analysis is required. However, if the Markov kernel P is ergodic then it is possible to obtain bounds in which the error constant arising in the analysis has milder growth with respect to J.*

- *Considering the case of deterministic dynamics shows just how difficult it may be to make the theorem applicable in practice: if the dynamics is deterministic then the original set of samples from μ_0, $\{v_0^{(n)}\}_{n=1}^N$ give rise to a set of particles $v_j^{(n)} = \Psi^{(j)}(v_0^{(n)})$; in other words the particle positions are* unaffected by the data*. This is clearly a highly undesirable situation, in general, since there is no reason at all why the pushforward under the dynamics of the initial measure μ_0 should have substantial overlap with the filtering distribution for a given fixed data set. Indeed for chaotic dynamical systems one would expect that it does not as the pushforward measure will be spread over the global attractor, whilst the data will, at fixed time, correspond to a single point on the attractor. This example motivates the improved proposals of the next section.*

♠

Before describing improvements to the basic particle filter, we prove the three lemmas underlying the convergence proof.

Lemma 4.7 *The sampling operator satisfies*

$$\sup_{\mu \in \mathcal{P}(\mathbb{R}^n)} d(S^N\mu, \mu) \le \frac{1}{\sqrt{N}}.$$

Proof Let ν be an element of $\mathcal{P}(\mathbb{R}^n)$ and $\{v^{(n)}\}_{n=1}^N$ i.i.d. with $v^{(1)} \sim \nu$. In this proof the randomness in the measure S^N arises from these samples $\{v^{(n)}\}_{n=1}^N$ and expectation over this randomness is denoted $\mathbb{E}$. Then

$$S^N\nu(f) = \frac{1}{N}\sum_{n=1}^{N} f\left(v^{(n)}\right)$$

and, defining $\overline{f} = f - \nu(f)$, we deduce that

$$S^N\nu(f) - \nu(f) = \frac{1}{N}\sum_{n=1}^{N} \overline{f}\left(v^{(n)}\right).$$

It is straightforward to see that

$$\mathbb{E}\overline{f}\left(v^{(n)}\right)\overline{f}\left(v^{(l)}\right) = \delta_{nl}\mathbb{E}\left|\overline{f}\left(v^{(n)}\right)\right|^2.$$

Furthermore, for $|f|_\infty \le 1$,

$$\mathbb{E}\left|\overline{f}\left(v^{(1)}\right)\right|^2 = \mathbb{E}\left|f\left(v^{(1)}\right)\right|^2 - \left|\mathbb{E}f\left(v^{(1)}\right)\right|^2 \le 1.$$

It follows that, for $|f|_\infty \le 1$,

$$\mathbb{E}\left|\nu(f) - S^N\nu(f)\right|^2 = \frac{1}{N^2}\sum_{n=1}^{N}\mathbb{E}\left|\overline{f}\left(v^{(n)}\right)\right|^2 \le \frac{1}{N}.$$

Since the result is independent of ν we may take the supremum over all probability measures and obtain the desired result. □

Lemma 4.8 *Since P is a Markov kernel we have*

$$d(P\nu, P\nu') \le d(\nu, \nu').$$

Proof Define

$$q(v') = \int_{\mathbb{R}^n} p(v', v)f(v)dv = \mathbb{E}(v_1|v_0 = v'),$$

that is the expected value of f under one-step of the Markov chain given by (2.1a), started from v'. Clearly, for $|f|_\infty \le 1$,

$$|q(v')| \le \int_{\mathbb{R}^n} p(v', dv)|f(v) \le \int_{\mathbb{R}^n} p(v', dv) = 1.$$

Thus

$$|q|_\infty \le \sup_v |q(v)| \le 1.$$

Note that

$$\begin{aligned}\nu(q) = \mathbb{E}(v_1|v_0 \sim \nu) &= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(v', v)f(v)\nu(dv')\,dv \\ &= \int_{\mathbb{R}^n}\Big(\int_{\mathbb{R}^n} p(v', v)\nu(dv')\Big)f(v)dv = P\nu(f).\end{aligned}$$

Thus $P\nu(f) = \nu(q)$ and it follows that

$$|P\nu(f) - P\nu'(f)| = |\nu(q) - \nu'(q)|.$$

Thus

$$\begin{aligned} d(P\nu, P\nu') &= \sup_{|f|_\infty \le 1} \Big(\mathbb{E}^\omega |P\nu(f) - P\nu'(f)|^2\Big)^{\frac{1}{2}} \\ &\le \sup_{|q|_\infty \le 1} \Big(\mathbb{E}^\omega |\nu(q) - \nu'(q)|^2\Big)^{\frac{1}{2}} \\ &= d(\nu, \nu') \end{aligned}$$

as required. □

Lemma 4.9 *Under the Assumptions of Theorem 4.5 we have*

$$d(L_j\nu, L_j\mu) \le 2\kappa^{-2} d(\nu, \mu).$$

Proof Notice that for $|f|_\infty < \infty$ we can rewrite

$$(L_j\nu)(f) - (L_j\mu)(f) = \frac{\nu(fg_j)}{\nu(g_j)} - \frac{\mu(fg_j)}{\mu(g_j)} \tag{4.32a}$$
$$= \frac{\nu(fg_j)}{\nu(g_j)} - \frac{\mu(fg_j)}{\nu(g_j)} + \frac{\mu(fg_j)}{\nu(g_j)} - \frac{\mu(fg_j)}{\mu(g_j)} \tag{4.32b}$$
$$= \frac{\kappa^{-1}}{\nu(g_j)}[\nu(\kappa f g_j) - \mu(\kappa f g_j)] + \frac{\mu(fg_j)}{\mu(g_j)}\frac{\kappa^{-1}}{\nu(g_j)}[\mu(\kappa g_j) - \nu(\kappa g_j)]. \tag{4.32c}$$

Now notice that $\nu(g_j)^{-1} \le \kappa^{-1}$ and that $\mu(fg_j)/\mu(g_j) \le 1$ since the expression corresponds to an expectation with respect to measure found from μ by reweighting with likelihood proportional to g_j. Thus

$$|(L_j\nu)(f) - (L_j\mu)(f)| \le \kappa^{-2}|\nu(\kappa f g_j) - \mu(\kappa f g_j)| + \kappa^{-2}|\nu(\kappa g_j) - \mu(\kappa g_j)|.$$

Since $|\kappa g_j|_\infty \le 1$ it follows that $|\kappa f g_j|_\infty$ and hence that

$$\mathbb{E}^\omega |(L_j\nu)(f) - (L_j\mu)(f)|^2 \le 4\kappa^{-4} \sup_{|h|_\infty \le 1} \mathbb{E}^\omega |\nu(h) - \mu(h)|^2.$$

The desired result follows. □

4.3.3. Improved Proposals

In the particle filter described in the previous section we propose according to the underlying unobserved dynamics, and then apply Bayes' formula to incorporate the data. The final point in Remarks 4.6 demonstrates that this may result in a very poor set of particles with which to approximate the filtering distribution. Cleverer proposals, which use the data, can lead to improved performance and we outline this methodology here.

Instead of moving the particles $\{v_j^{(n)}\}_{n=1}^N$ according to the Markov kernel P, we use a Markov kernel Q_j with density $\mathbb{Q}(v_{j+1}|v_j, Y_{j+1})$. The weights $w_{j+1}^{(n)}$ are found, as before, by applying Bayes' formula for each particle, and then weighting appropriately as in (4.27):

$$\begin{aligned} \widetilde{w}_{j+1}^{(n)} &= w_j^{(n)} \frac{\mathbb{P}\left(y_{j+1}|\widehat{v}_{j+1}^{(n)}\right)\mathbb{P}\left(\widehat{v}_{j+1}^{(n)}|v_j^{(n)}\right)}{\mathbb{Q}\left(\widehat{v}_{j+1}^{(n)}|v_j^{(n)}, Y_{j+1}\right)}, && (4.33a)\\ w_{j+1}^{(n)} &= \widetilde{w}_{j+1}^{(n)} \Big/ \left(\sum_{n=1}^{N} \widetilde{w}_{j+1}^{(n)}\right). && (4.33b) \end{aligned}$$

The choice

$$\mathbb{Q}\left(v_{j+1}|v_j^{(n)}, Y_{j+1}\right) = \mathbb{P}\left(v_{j+1}|v_j^{(n)}\right)$$

results in the bootstrap filter from the preceding subsection. In the more general case the approach results in the following algorithm:

1. Set $j = 0$ and $\mu_0^N(v_0)dv_0 = \mathbb{P}(v_0)dv_0$.
2. Draw $v_j^{(n)} \sim \mu_j^N$, $n = 1, \dots, N$.
3. Set $w_j^{(n)} = 1/N$, $n = 1, \dots, N$.
4. Draw $\widehat{v}_{j+1}^{(n)} \sim \mathbb{Q}(\cdot|v_{j+1}^{(n)}, Y_{j+1})$.
5. Define $w_{j+1}^{(n)}$ by (4.33) and $\mu_{j+1}^N = \mathbb{P}^N(v_{j+1}|Y_{j+1})$ by (4.26).
6. $j + 1 \to j$.
7. Go to step 2.

We note that the normalization constants in (4.33a), here assumed known in the definition of the reweighting, or not of course needed. The so-called **optimal proposal** is found by choosing

$$\mathbb{Q}\left(v_{j+1}|v_j^{(n)}, Y_{j+1}\right) \equiv \mathbb{P}\left(v_{j+1}|v_j^{(n)}, y_{j+1}\right)$$

which results in

$$\widetilde{w}_{j+1}^{(n)} = w_j^{(n)} \mathbb{P}\left(y_{j+1}|v_j^{(n)}\right). \tag{4.34}$$

The above can be seen by observing that the definition of conditional probability gives

$$\begin{aligned} \mathbb{P}\left(y_{j+1}|\widehat{v}_{j+1}^{(n)}\right)\mathbb{P}\left(\widehat{v}_{j+1}^{(n)}|v_j^{(n)}\right) &= \mathbb{P}\left(y_{j+1}, \widehat{v}_{j+1}^{(n)}|v_j^{(n)}\right)\\ &= \mathbb{P}\left(\widehat{v}_{j+1}^{(n)}|v_j^{(n)}, y_{j+1}\right)\mathbb{P}\left(y_{j+1}|v_j^{(n)}\right). \end{aligned} \tag{4.35}$$

Substituting the optimal proposal into (4.33) then immediately gives (4.34).

This small difference from the bootstrap filter may seem trivial at a glance, and at the potentially large cost of sampling from $\mathbb{Q}$. However, in the case of nonlinear Gaussian Markov models as we study here, the distribution and the weights are given in closed form. If the dynamics is highly nonlinear or the model noise is larger than the observational noise then the variance of the weights for the optimal proposal may be much smaller than for the standard proposal. The corresponding particle filter will be referred to with the acronym SIRS(OP) to indicate the optimal proposal . For deterministic dynamics the optimal proposal reduces to the standard proposal.

4.4 Large-Time Behaviour of Filters

With the exception of the Kalman filter for linear problems, and the particle filter in the general case, the filtering methods presented in this chapter do not, in general, give accurate approximations of the true posterior distribution; in particular the approximate Gaussian filters do not perform well as measured by the Bayesian quality assessment test of section 2.7. However they may perform well as measured by the signal estimation quality assessment test and the purpose of this section is to demonstrate this fact.

More generally, an important question concerning filters is their behaviour when iterated over long times and, in particular, their ability to recover the true signal underlying the data if iterated for long enough, even when initialized far from the truth. In this section we present some basic large time asymptotic results for filters to illustrate the key issue which affects the ability of filters to accurately recover the signal when iterated for long enough. The main idea is that the data must be sufficiently rich to stabilize any inherent instabilities within the underlying dynamical model (2.1); in rough terms it is necessary to observe only the unstable directions as the dynamics of the model itself will enable recovery of the true signal within the space spanned by the stable directions. We illustrate this idea first, in subsection 4.4.1, for the explicitly solvable case of the Kalman filter in one dimension, and then, in subsection 4.4.2, for the 3DVAR method.

4.4.1. The Kalman Filter in One Dimension

We consider the case of one dimensional dynamics with

$$\Psi(v) = \lambda v, \quad h(v) = v,$$

while we will also assume that

$$\Sigma = \sigma^2, \quad \Gamma = \gamma^2.$$

With these definitions equations (4.3a,b) become

$$\frac{1}{c_{j+1}} = \frac{1}{\sigma^2 + \lambda^2 c_j} + \frac{1}{\gamma^2}, \tag{4.36a}$$

$$\frac{m_{j+1}}{c_{j+1}} = \frac{\lambda m_j}{\sigma^2 + \lambda^2 c_j} + \frac{1}{\gamma^2} y_{j+1}, \tag{4.36b}$$

which, after some algebraic manipulations, give

$$c_{j+1} = g(c_j), \tag{4.37a}$$

$$m_{j+1} = \left(1 - \frac{c_{j+1}}{\gamma^2}\right) \lambda m_j + \frac{c_{j+1}}{\gamma^2} y_{j+1}, \tag{4.37b}$$

where we have defined

$$g(c) := \frac{\gamma^2(\lambda^2 c + \sigma^2)}{\gamma^2 + \lambda^2 c + \sigma^2}. \tag{4.38}$$

We wish to study the behaviour of the Kalman filter as $j \to \infty$, *i.e.* when more and more data points are assimilated into the model. Note that the covariance evolves independently of the data $\{y_j\}_{j \in \mathbb{Z}^+}$ and satisfies an autonomous nonlinear dynamical system. However it is of interest to note that, if $\sigma^2 = 0$, then the dynamical system for c_j^{-1} is linear.

We now study the asymptotic properties of this map. The fixed points $c^\star$ of (4.37a) satisfy

$$c^\star = \frac{\gamma^2(\lambda^2 c^\star + \sigma^2)}{\gamma^2 + \lambda^2 c^\star + \sigma^2}, \tag{4.39}$$

and thus solve the quadratic equation

$$\lambda^2(c^\star)^2 + \big(\gamma^2(1-\lambda^2) + \sigma^2\big)c^\star - \gamma^2\sigma^2 = 0.$$

We see that, provided $\lambda\gamma\sigma \neq 0$, one root is positive and one negative. The roots are given by

$$c^\star_\pm = \frac{-(\gamma^2+\sigma^2-\gamma^2\lambda^2) \pm \sqrt{(\gamma^2+\sigma^2-\gamma^2\lambda^2)^2 + 4\lambda^2\gamma^2\sigma^2}}{2\lambda^2}. \tag{4.40}$$

We observe that the update formula for the covariance ensures that, provided $c_0 \geq 0$ then $c_j \geq 0$ for all $j \in \mathbb{N}$. It also demonstrates that $c_j \leq \gamma^2$ for all $j \in \mathbb{Z}^+$ so that the variance of the filter is no larger than the variance in the data. We may hence fix our attention on non-negative covariances, knowing that they are also uniformly bounded by γ^2. We will now study the stability of the non-negative fixed points .

We first start with the case $\sigma = 0$, which corresponds to deterministic dynamics, and for which the dynamics of c_j^{-1} is linear. In this case we obtain

$$c^\star_+ = 0, \quad c^\star_- = \frac{\gamma^2(\lambda^2-1)}{\lambda^2},$$

and

$$g'(c^\star_+) = \lambda^2, \quad g'(c^\star_-) = \lambda^{-2},$$

which implies that when $\lambda^2 < 1$, $c^\star_+$ is an asymptotically stable fixed point, while when $\lambda^2 > 1$, $c^\star_-$ is an asymptotically stable fixed point . When $|\lambda| = 1$ the two roots are coincident at the origin and neutrally stable. Using the aforementioned linearity, for the case $\sigma = 0$ it is possible to solve (4.36a) to obtain for $\lambda^2 \neq 1$

$$\frac{1}{c_j} = \left(\frac{1}{\lambda^2}\right)^j \frac{1}{c_0} + \frac{1}{\gamma^2}\left[\frac{\left(\frac{1}{\lambda^2}\right)^j - 1}{\frac{1}{\lambda^2} - 1}\right]. \tag{4.41}$$

This explicit formula shows that the fixed point $c^\star_+$ (resp. $c^\star_-$) is globally asymptotically stable, and exponentially attracting on $\mathbb{R}^+$, when $\lambda^2 < 1$ (resp. $\lambda^2 > 1$). Notice also that $c^\star_- = \mathcal{O}(\gamma^2)$ so that when $\lambda^2 > 1$, the asymptotic variance of the filter scales as the observational noise variance. Furthermore, when $\lambda^2 = 1$ we may solve (4.36a) to obtain

$$\frac{1}{c_j} = \frac{1}{c_0} + \frac{j}{\gamma^2},$$

showing that $c^\star_- = c^\star_+ = 0$ is globally asymptotically stable on $\mathbb{R}^+$, but is only algebraically attracting.

We now study the stability of the fixed points $c^\star_+$ and $c^\star_-$ in the case of $\sigma^2 > 0$ corresponding to the case where the dynamics are stochastic. To this end we prove some bounds on $g'(c^\star)$ that will also be useful when we study the behaviour of the error between the true signal and the estimated mean; here, and in what follows in the remainder of this example, prime denotes differentiation with respect to c. We start by noting that

$$g(c) = \gamma^2 - \frac{\gamma^4}{\gamma^2 + \lambda^2 c + \sigma^2}, \tag{4.42}$$

and so

$$g'(c) = \frac{\lambda^2\gamma^4}{(\gamma^2 + \lambda^2 c + \sigma^2)^2}.$$

Using the fact that $c^\star$ satisfies (4.39) together with equation (4.42), we obtain

$$g'(c^\star) = \frac{1}{\lambda^2}\frac{(c^\star)^2}{\left(c^\star + \frac{\sigma^2}{\lambda^2}\right)^2} \quad \text{and} \quad g'(c^\star) = \lambda^2\left(1 - \frac{c^\star}{\gamma^2}\right)^2.$$

We can now see that from the first equation we obtain the following two bounds, since $\sigma^2 > 0$:

$$g'(c^\star) < \lambda^{-2}, \quad \text{for} \quad \lambda \in \mathbb{R}, \quad \text{and} \quad g'(c^\star) < 1, \quad \text{for} \quad \lambda^2 = 1,$$

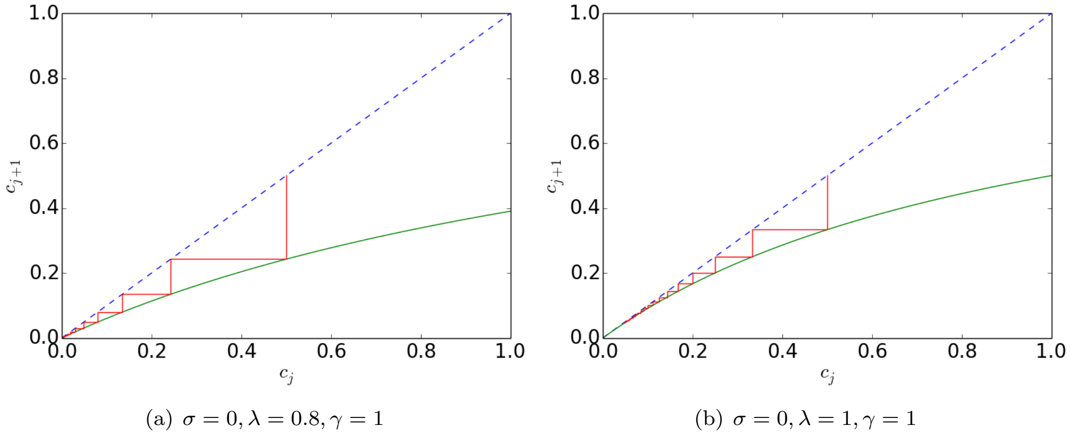
while from the second equality and the fact that, since $c^\star$ satisfies (4.39), $c^\star < \gamma^2$ we obtain

$$g'(c^\star) < \lambda^2.$$

when $c^\star > 0$. We thus conclude that when $\sigma^2 > 0$ the fixed point $c_+^\star$ of (4.37a) is always stable independently of the value of the parameter λ.

	Limiting covariance for $\sigma^2 = 0$	Limiting covariance for $\sigma^2 > 0$
$\lvert\lambda\rvert < 1$	$c_j \to 0$ (exponentially)	$c_j \to c_+^\star = \mathcal{O}(\gamma^2)$ (exponentially)
$\lvert\lambda\rvert = 1$	$c_j \to 0$ (algebraically)	$c_j \to c_+^\star = \mathcal{O}(\gamma^2)$ (exponentially)
$\lvert\lambda\rvert > 1$	$c_j \to c_-^\star = \mathcal{O}(\gamma^2)$ (exponentially)	$c_j \to c_+^\star = \mathcal{O}(\gamma^2)$ (exponentially)

Table 4.1: Summary of the limiting behaviour of covariance c_j for Kalman filter applied to one dimensional dynamics

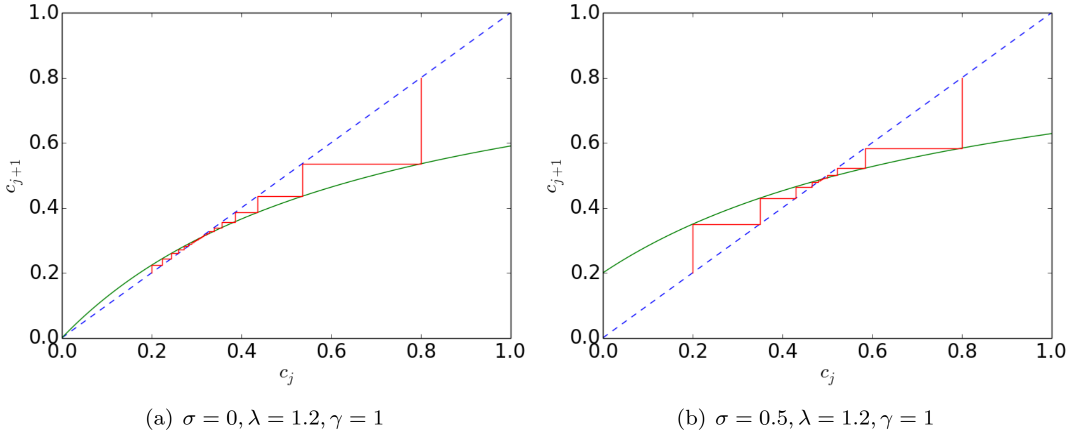


(a) $\sigma = 0, \lambda = 0.8, \gamma = 1$

(b) $\sigma = 0, \lambda = 1, \gamma = 1$

Figure 4.1: Cobweb diagram for equation (4.37a)

Table 4.1 summarises the behaviour of the variance of the Kalman filter in the case of one-dimensional dynamics. This is illustrated further in Figures 4.1 and 4.2 where we plot the cobweb diagram for the map (4.42). In particular, in Figure 4.1 we observe the difference between the algebraic and geometric convergence to 0, for different values of λ in the case $\sigma = 0$, while in Figure 4.2 we observe the exponential convergence to $c_+^\star$ for the case of $|\lambda| > 1$. The analysis of the error between the mean and the truth underlying the data is left as an exercise at the end of the chapter. This shows that the error in the mean is, asymptotically, of order γ^2 in the case where $\sigma = 0$.



(a) $\sigma = 0, \lambda = 1.2, \gamma = 1$

(b) $\sigma = 0.5, \lambda = 1.2, \gamma = 1$

Figure 4.2: Cobweb diagram for equation (4.37a)

4.4.2. The 3DVAR Filter

In the previous subsection we showed that the Kalman filter accurately recovers any one-dimensional signal, provided the observational noise is small. The result allows for initialization far from the true signal and is, in this sense, quite strong. On the other hand being only one-dimensional it gives a somewhat limited picture. In this subsection we study the 3DVAR filter given by (4.14). We study conditions under which the 3DVAR filter will recover the true signal, to within a small observational noise level of accuracy, in dimensions bigger than one, and when only part of the system is observed.

To this end we assume that

$$y_{j+1} = Hv^\dagger_{j+1} + \epsilon_j \tag{4.43}$$

where the true signal $\{v^\dagger_j\}_{j\in\mathbb{N}}$ satisfies

$$v^\dagger_{j+1} = \Psi(v^\dagger_j),\ j \in \mathbb{N} \tag{4.44a}$$

$$v^\dagger_0 = u \tag{4.44b}$$

and, for simplicity, we assume that the observational noise satisfies

$$\sup_{j\in\mathbb{N}} |\epsilon_j| = \epsilon. \tag{4.45}$$

We have the following result.

Theorem 4.10 *Assume that the data is given by* (4.43)*, where the signal follows equation* (4.44) *and the error in the data satisfies* (4.45)*. Assume furthermore that* $\widehat{C}$ *is chosen so that* $(I - KH)\Psi : \mathbb{R}^n \to \mathbb{R}^n$ *is globally Lipschitz with constant* $a < 1$ *in some norm* $\|\cdot\|$*. Then there is constant* $c > 0$ *such that*

$$\limsup_{j\to\infty} \|m_j - v^\dagger_j\| \le \frac{c}{1-a}\epsilon.$$

Proof We may write (4.14), (4.44), using (4.43), as

$$m_{j+1} = (I - KH)\Psi(m_j) + KH\Psi(v^\dagger_j) + K\epsilon_j$$
$$v^\dagger_{j+1} = (I - KH)\Psi(v^\dagger_j) + KH\Psi(v^\dagger_j).$$

Subtracting, and letting $e_j = m_j - v_j^\dagger$ gives, for some finite constant c independent of j,

$$\begin{aligned}\|e_{j+1}\| &\le \|(I-KH)\Psi(m_j) - (I-KH)\Psi(v_j^\dagger)\| + \|K\epsilon_j\| \\ &\le a\|e_j\| + c\epsilon.\end{aligned}$$

Applying the Gronwall Lemma 1.14 gives the desired result. □

We refer to a map with Lipschitz constant less than 1 as a *contraction* in what follows.

Remark 4.11 *The preceding simple theorem shows that it is possible to construct filters which can recover from being initialized far from the truth and lock-on to a small neighbourhood of the true signal underlying the data, when run for long enough. Furthermore, this can happen even when the system is only partially observed, provided that the observational noise is small and enough of the system is observed. This concept of observing "enough" illustrates a key idea in filtering: the question of whether the fixed model covariance in 3DVAR, $\widehat{C}$, can be chosen to make $(I-KH)\Psi$ into a contraction involves a subtle interplay between the underlying dynamics, encapsulated in Ψ, and the observation operator H. In rough terms the question of making $(I-KH)\Psi$ into a contraction is the question of whether the unstable parts of the dynamics are observed; if they are then it is typically the case that $\widehat{C}$ can be designed to obtain the desired contraction.* ♠

Example 4.12 *Assume that $H = I$, so that the whole system is observed, that $\Gamma = \gamma^2 I$ and $\widehat{C} = \sigma^2 I$. Then, for $\eta^2 = \frac{\gamma^2}{\sigma^2}$*

$$S = (\sigma^2+\gamma^2)I, \quad K = \frac{\sigma^2}{(\sigma^2+\gamma^2)}I$$

and

$$(I-KH) = \frac{\gamma^2}{(\sigma^2+\gamma^2)}I = \frac{\eta^2}{(1+\eta^2)}I.$$

Thus, if $\Psi : \mathbb{R}^n \to \mathbb{R}^n$ is globally Lipschitz with constant $\lambda > 0$ in the Euclidean norm, $|\cdot|$, then $(I-KH)\Psi$ is globally Lipschitz with constant $a < 1$, if η is chosen so that $\frac{\eta^2\lambda}{1+\eta^2} < 1$. Thus, by choosing η sufficiently small the filter can be made to contract. This corresponds to trusting the data sufficiently in comparison to the model. It is a form of **variance inflation** *in that, for given level of observational noise, η can be made sufficiently small by choosing the model variance scale σ^2 sufficiently large – "inflating" the model variance.* ♠

Example 4.13 *Assume that there is a partition of the state space in which $H = (I, 0)^T$, so that only part of the system is observed. Set $\Gamma = \gamma^2 I$ and $\widehat{C} = \sigma^2 I$. Then, with η as in the previous example,*

$$I - KH = \begin{pmatrix} \frac{\eta^2}{1+\eta^2}I & 0 \\ 0 & I \end{pmatrix}.$$

Whilst the previous example shows that variance inflation may help to stabilize the filter, this example shows that, in general, more is required: in this case it is clear that making $(I-KH)\Psi(\cdot)$ into a contraction will require a relationship between the subspace in which we observe and the space in which the dynamics of the map is expanding and contracting. For example, if $\Psi(u) = Lu$ and

$$L = \begin{pmatrix} 2I & 0 \\ 0 & aI \end{pmatrix}$$

then

$$(I - KH)L = \begin{pmatrix} \frac{2\eta^2}{1+\eta^2}I & 0 \\ 0 & aI \end{pmatrix}$$

When $|a| < 1$ *this can be made into a contraction by choosing* η *sufficiently small; but for* $|a| \geq 1$ *this is no longer possible. The example thus illustrates the intuitive idea that the observations should be sufficiently rich to ensure that the unstable directions within the dynamics can be tamed by observing them.*

♠

4.4.3. The Synchronization Filter

A fundamental idea underlying successful filtering of partially observed dynamical systems, is synchronization. To illustrate this we introduce and study the idealized *synchronization filter*. To this end consider a partition of the identity $P + Q = I$. We write $v = (p, q)$ where $p = Pv, q = Qv$ and then, with a slight abuse of notation, write $\Psi(v) = \Psi(p, q)$. Consider a true signal governed by the deterministic dynamics model (4.44) and write $v_k^\dagger = (p_k^\dagger, q_k^\dagger)$, with $p_k^\dagger = Pv_k^\dagger$ and $q_k^\dagger = Qv_k^\dagger$. Then

$$\begin{aligned} p_{k+1}^\dagger &= P\Psi(p_k^\dagger, q_k^\dagger), \\ q_{k+1}^\dagger &= Q\Psi(p_k^\dagger, q_k^\dagger). \end{aligned}$$

Now imagine that we observe $y_k = p_k^\dagger$ exactly, without noise. Then the synchronization filter simply fixes the image under P to $p_k^\dagger$ and plugs this into the image of the dynamical model under Q; if the filter is $m_k = (p_k, q_k)$ with $p_k = Pm_k$ and $q_k = Qm_k$ then

$$\begin{aligned} p_{k+1} &= p_{k+1}^\dagger, \\ q_{k+1} &= Q\Psi(p_k^\dagger, q_k). \end{aligned}$$

We note that, expressed in terms of the data, this filter has the form

$$m_{k+1} = Q\Psi(m_k) + Py_{k+1}. \tag{4.46}$$

A key question now is whether or not the filter synchronizes in the following sense:

$$|q_k - q_k^\dagger| \to 0 \text{ as } k \to \infty.$$

This of course is equivalent to

$$|m_k - v_k^\dagger| \to 0 \text{ as } k \to \infty. \tag{4.47}$$

Whether or not this happens involves, as for 3DVAR described above, a subtle interplay between the underlying dynamics and the observation operator, here P. The bibliography section 4.6 contains pointers to the literature studying this question.

In fact the following example shows how the synchronization filter can be viewed as a distinguished parameter limit, corresponding to infinite variance inflation, for a particular family of 3DVAR filters.

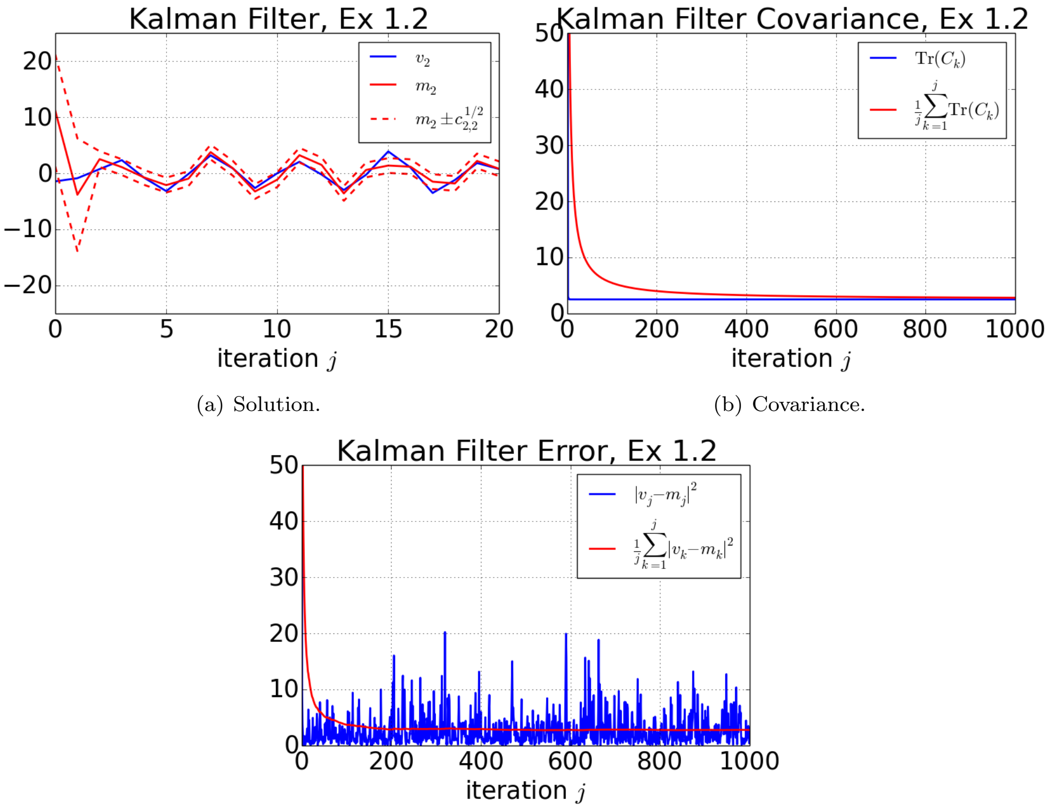
Example 4.14 *Let $H = P$ and $\Gamma = \gamma^2 I$. If we choose $\widehat{C}$ as in Example 4.13 then the 3DVAR filter can be written as*

$$m_{k+1} = S\Psi(m_k) + (I - S)y_{k+1}, \tag{4.48a}$$

$$S = \frac{\eta^2}{1+\eta^2}P + Q. \tag{4.48b}$$

The limit $\eta \to 0$ is the the extreme limit of variance inflation referred to in Example 4.12. In this limit the 3DVAR filter becomes the synchronization filter (4.46). ♠

4.5 Illustrations



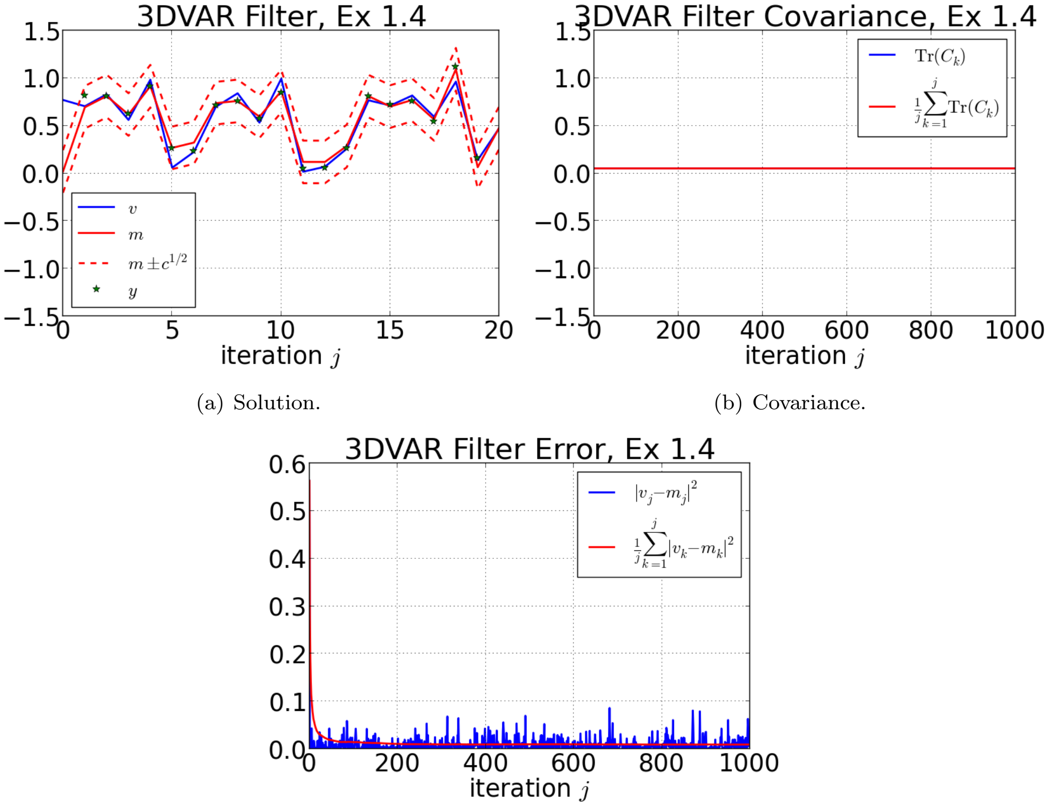
(a) Solution.

(b) Covariance.

(c) Error.

Figure 4.3: Kalman filter applied to the linear system of Example 2.2 with $A = A_3$, $H = (1, 0)$, $\Sigma = I$, and $\Gamma = 1$, see also `p8.m` in Section 5.2.5. The problem is initialized with mean 0 and covariance $10\,I$.

The first illustration concerns the Kalman filter applied to the linear system of Example 2.2 with $A = A_3$. We assume that $H = (1, 0)$ so that we observe only the first component of the system and the model and observational covariances are $\Sigma = I$ and $\Gamma = 1$, where I is the 2×2 identity. The problem is initialized with mean 0 and covariance $10\,I$. Figure 4.3a shows the behaviour of the filter on the unobserved component, showing how the mean locks onto



(a) Solution.

(b) Covariance.

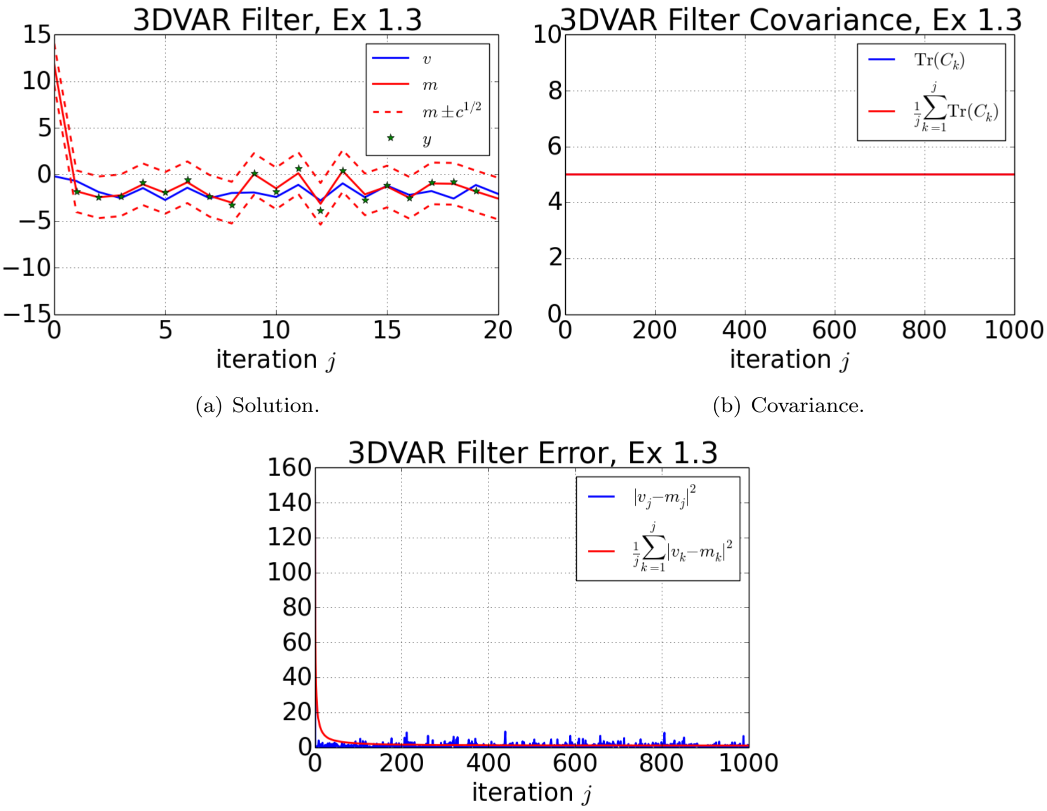
(c) Error.

Figure 4.4: 3DVAR methodology applied to the logistic map Example 2.4 with $r = 4$, $\gamma^2 = 10^{-2}$, and $c = \gamma^2/\eta$ with $\eta = 0.2$, see also `p9.m` in section 5.3.1.

a small neighbourhood of the truth and how the one-standard deviation confidence intervals computed from the variance on the second component also shrink from a large initial value to an asymptotic small value; this value is determined by the observational noise variance in the first component. In Figure 4.3b the trace of the covariance matrix is plotted demonstrating that the total covariance matrix asymptotes to a small limiting matrix. And finally Figure 4.3c shows the error (in the Euclidean norm) between the filter mean and the truth underlying the data, together with its running average. We will employ similar figures (a), (b) and (c) in the examples which follow in this section.

The next illustration shows the 3DVAR algorithm applied to the Example 2.4 with $r = 2.5$. We consider noise-free dynamics and observational variance of $\gamma^2 = 10^{-2}$. The fixed model covariance is chosen to be $c = \gamma^2/\eta$ with $\eta = 0.2$. The resulting algorithm performs well at tracking the truth with asymptotic time-averaged Euclidean error of size roughly 10^{-2}. See Figure 4.4.

The rest of the figures illustrate the behaviour of the various filters, all applied to the Example 2.3 with $\alpha = 2.5, \sigma = 0.3$, and $\gamma = 1$. In particular, 3DVAR(Figure 4.5), ExKF(Figure 4.6), EnKF (Figure 4.7), ETKF (Figure 4.8), and the particle filter with standard (Figure 4.9) and optimal (Figure 4.10) proposals are all compared on the same example. The ensemble-based methods all use 100 ensemble members each (notice this is much larger than



(a) Solution.

(b) Covariance.

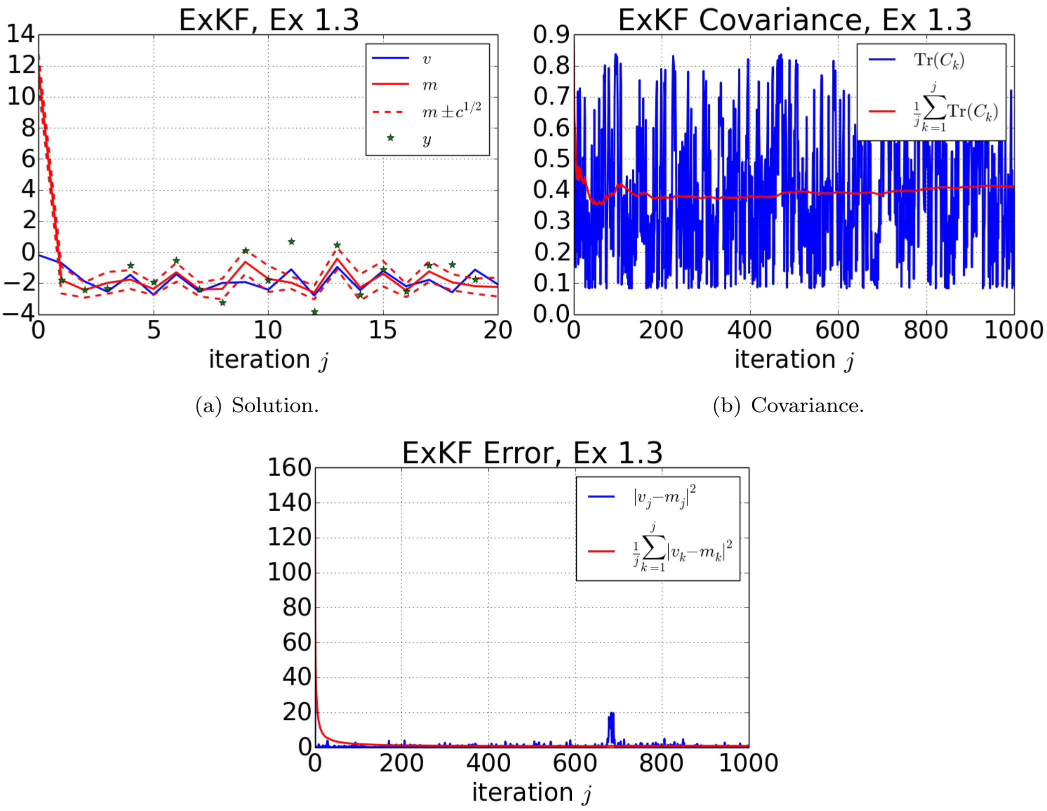
(c) Error.

Figure 4.5: 3DVAR for the sin map Example 2.3 with $\alpha = 2.5$, $\sigma = 0.3, \gamma = 1$, and $\eta = 0.2$, see also `p10.m` in Section 5.3.3.

the dimension of the state space which is $n = 1$ here, and so a regime outside of which the ensemble methods would usually be employed in practice). For 3DVAR, results from which (for this example) are only shown in the summary Figure 4.11, we take $\eta = 0.5$.

All of the methods perform well at tracking the true signal, asymptotically in time, recovering from a large initial error. However they also all exhibit occasional instabilities, and lose track of the true signal for short periods of time. From Fig. 4.6(c) we can observe that the ExKF has small error for most of the simulation, but that sporadic large excursions are seen in the error. From Fig. 4.8(c) one can observe that ETKF is similarly prone to small destabilization and local instability as the EnKF with perturbed observations in Fig. 4.7(c). Also, notice from Figure 4.9(c) that the particle filter with standard proposal is perhaps slightly more prone to destabilization than the optimal proposal in Figure 4.10(c), although the difference is minimal.

The performance of the filters is now compared through a detailed study of the statistical properties of the error $e = m - v^\dagger$, over long simulation times. In particular we compare the histograms of the errors, and their large time averages. Figure 4.11 compares the errors incurred by the three basic methods 3DVAR , ExKF, and EnKF, demonstrating that the EnKF is the most accurate method of the three on average, with ExKF the least accurate on average. Notice from Fig. 4.11(a) that the error distribution of 3DVAR is the widest,



(a) Solution.

(b) Covariance.

(c) Error.

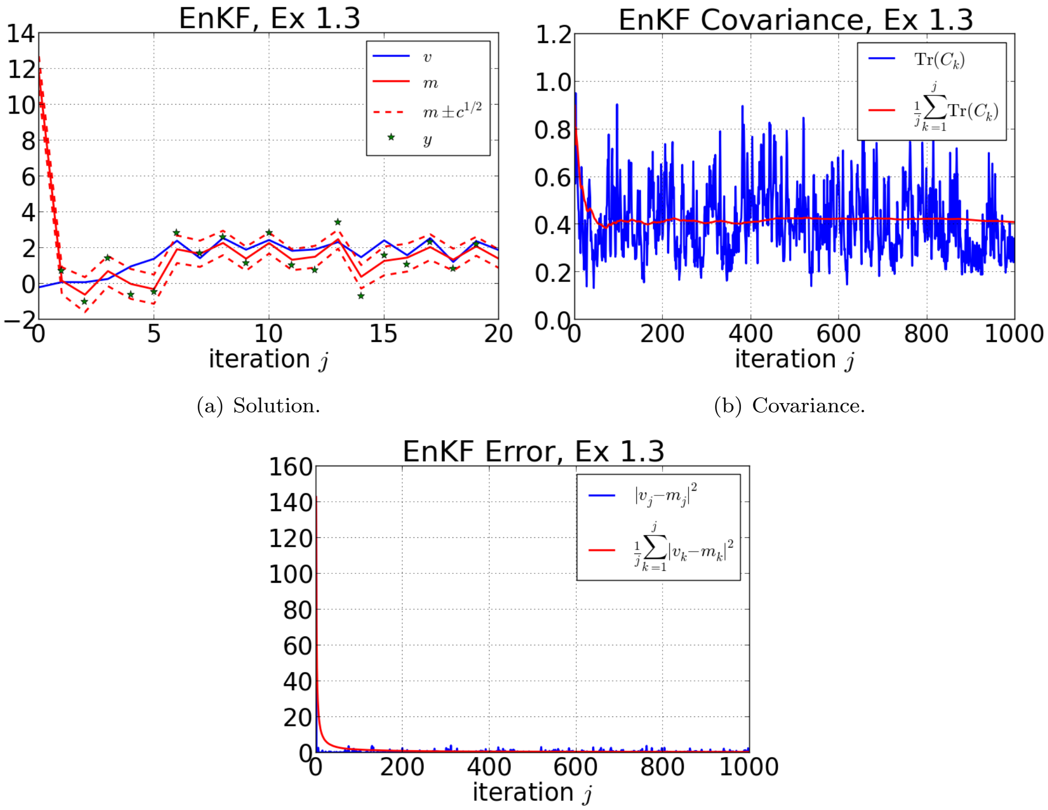
Figure 4.6: ExKF on the sin map Example 2.3 with $\alpha = 2.5$, $\sigma = 0.3$, and $\gamma = 1$, see also `p11.m` in Section 5.3.4.

and both it and EnKF remain consistently accurate. The distribution of ExKF is similar to EnKF, except with "fat tails" associated to the destabilization intervals seen in Fig. 4.6.

Figure 4.12 compares the errors incurred by the four more accurate ensemble-based methods EnKF, ETKF, SIRS, and SIRS(OP). The error distribution, Fig. 4.12(a) of all these filters is similar. In Fig. 4.12(b) one can see that the time-averaged error is indistinguishable between EnKF and ETKF. Also, the EnKF, ETKF, and SIRS(OP) also remain more or less consistently accurate. The distribution of e for SIRS is similar to SIRS(OP), except with fat tails associated to the destabilization intervals seen in Fig. 4.9, which leads to the larger time-averaged error seen in Fig. 4.12(b). In this sense, the distribution of e is similar to that for ExKF.

4.6 Bibliographic Notes

- Section 4.1 The Kalman Filter has found wide-ranging application to low dimensional engineering applications where the linear Gaussian model is appropriate, since its introduction in 1960 [65]. In addition to the original motivation in control of flight vehicles, it has grown in importance in the fields of econometric time-series analysis, and signal



(a) Solution.

(b) Covariance.

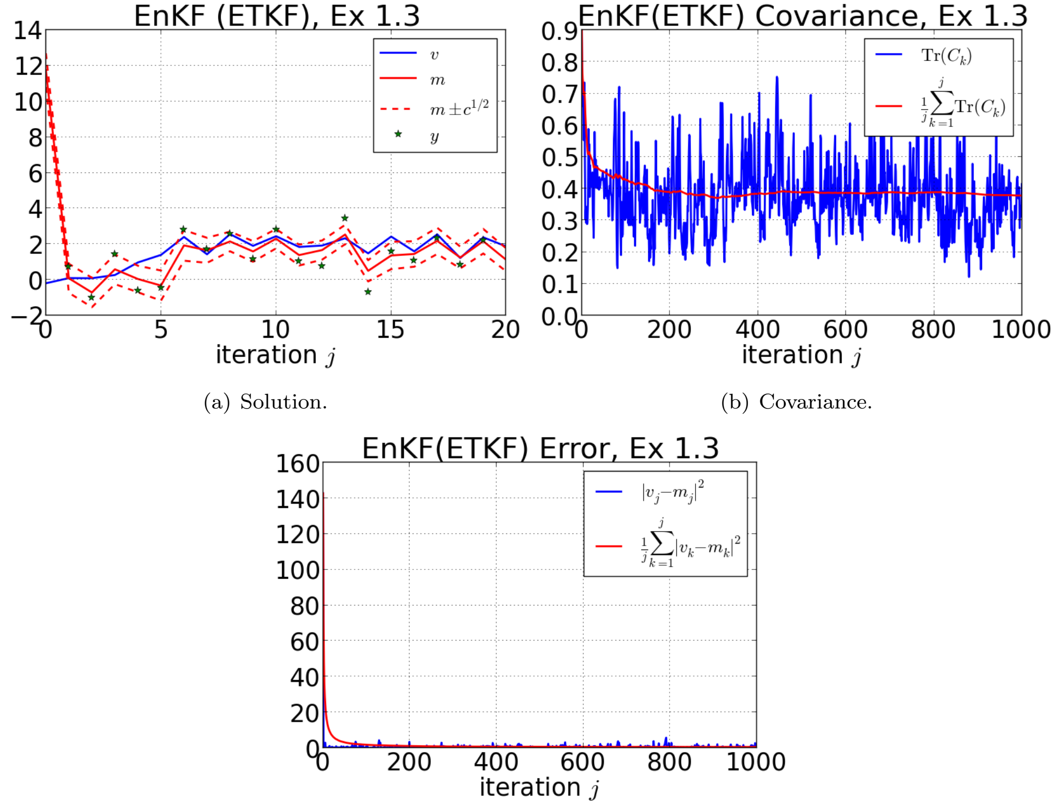
(c) Error.

Figure 4.7: EnKF on the sin map Example 2.3 with $\alpha = 2.5$, $\sigma = 0.3$, $\gamma = 1$ and $N = 100$, see also `p12.m` in Section 5.3.5.

processing [52]. It is also important because it plays a key role in the development of the approximate Gaussian filters which are the subject of section 4.2. The idea behind the Kalman filter, to optimally combine model and data, is arguably one of the most important ideas in applied mathematics over the last century: the impact of the paper [65] on many applications domains has been huge.

- Section 4.2 All the non-Gaussian Filters we discuss are based on modifying the Kalman filter so that it may be applied to non-linear problems. The development of new filters is a very active area of research and the reader is directed to the book [82], together with the articles [25],[83] and [114] for insight into some of the recent developments with an applied mathematics perspective.

  The 3DVAR algorithm was proposed at the UK Met Office in 1986 [75, 76], and was subsequently developed by the US National Oceanic and Atmospheric Administration [95] and by the European Centre for Medium-Range Weather Forecasts (ECMWF) in [32]. The perspective of these papers was one of minimization and, as such, easily incorporates nonlinear observation operators via the objective functional (4.11), with a fixed $\widehat{C} = \widehat{C}_{j+1}$, for the analysis step of filtering; nonlinear observation operators are important in numerous applications, including numerical weather forecasting. In



(a) Solution.

(b) Covariance.

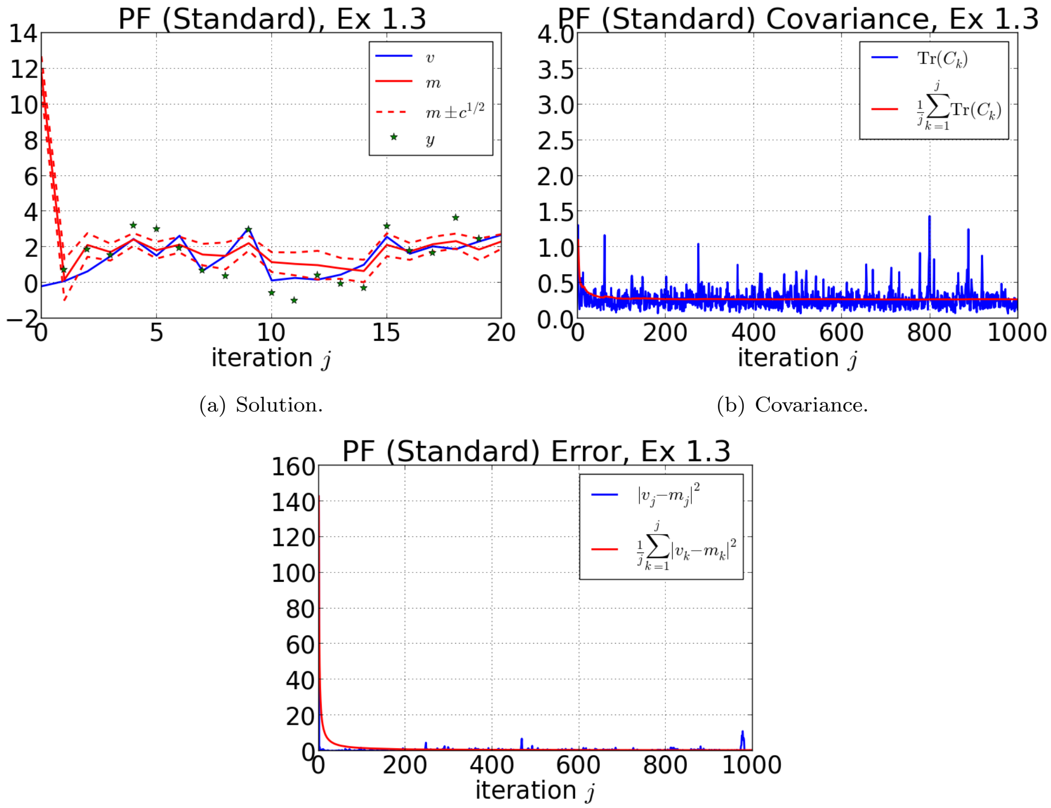
(c) Error.

Figure 4.8: ETKF on the sin map Example 2.3 with $\alpha = 2.5$, $\sigma = 0.3$, $\gamma = 1$ and $N = 100$, see also `p13.m` in Section 5.3.6.

the case of linear observation operators the objective functional is given by (4.12) with explicit solution given, in the case $\widehat{C} = \widehat{C}_{j+1}$, by (4.14). In fact the method of *optimal interpolation* predates 3DVAR and takes the linear equations (4.14) as the starting point, rather than starting from a minimization principle; it is then very closely related to the method of krigging from the geosciences [108]. The 3DVAR algorithm is important because it is prototypical of the many more sophisticated filters which are now widely used in practice and it is thus natural to study it.

The extended Kalman filter was developed in the control theory community and is discussed at length in [63]. It is not practical to implement in high dimensions, and low-rank extended Kalman filters are then used instead; see [71] for a recent discussion.

The ensemble Kalman filter uses a set of particles to estimate covariance information, and may be viewed as an approximation of the extended Kalman filter, designed to be suitable in high dimensions. See [40] for an overview of the methodology, written by one of its originators, and [116] for an early example of the power of the method. We note that the minimization principle (4.15) has the very desirable property that the samples $\{\widehat{v}_{n+1}^{(n)}\}_{n=1}^{N}$ correspond, to samples of the Gaussian distribution found by Bayes theorem with prior $N(\widehat{m}_{j+1}, \widehat{C}_{j+1})$ likelihood $y_{j+1}|v$. This is the idea behind the



(a) Solution.

(b) Covariance.

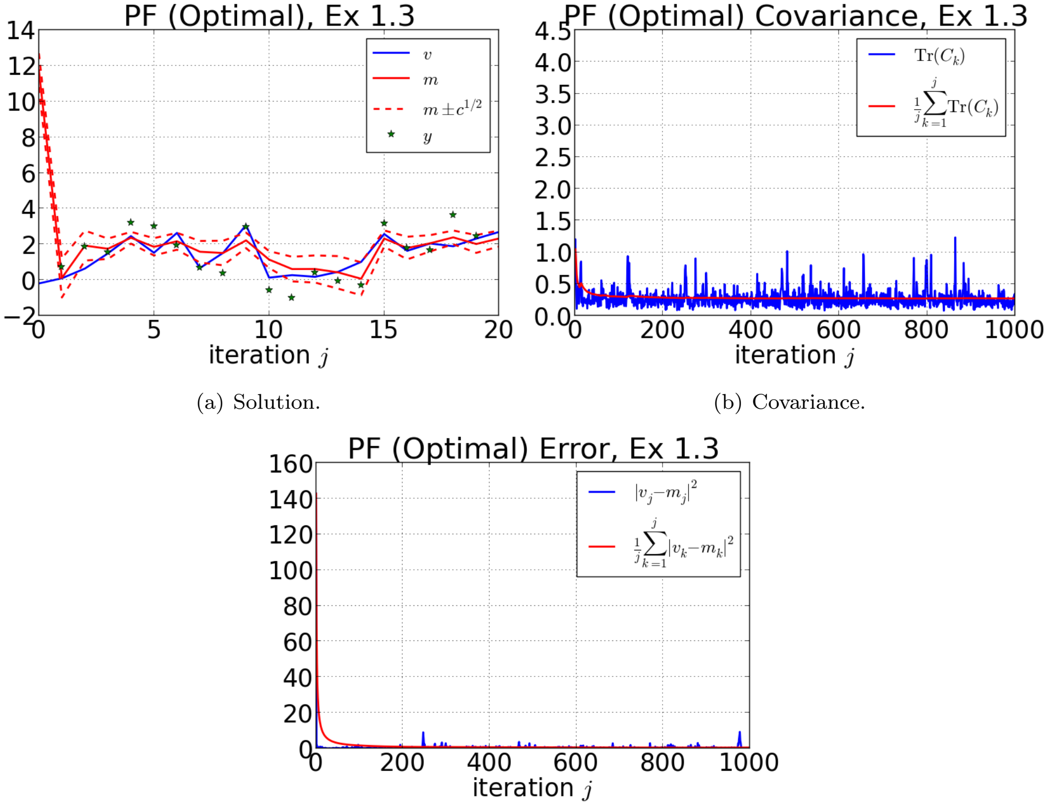
(c) Error.

Figure 4.9: Particle Filter (standard proposal) on the sin map Example 2.3 with $\alpha = 2.5$, $\sigma = 0.3$, $\gamma = 1$ and $N = 100$, see also `p14.m` in Section 5.3.7.

randomized maximum likelihood method described in [93], and widely used in petroleum applications; the idea is discussed in detail in the context of the EnKF in [67]. There has been some analysis of the EnKF in the large sample limit; see for example [73, 72, 85]. However, the primary power of the method for practitioners is that it seems to provide useful information for small sample sizes; it is therefore perhaps a more interesting direction for analysis to study the behaviour of the algorithm, and determine methodologies to improve it, for fixed numbers of ensemble members. There is some initial work in this direction and we describe it below.

Note that the Γ appearing in the perturbed observation EnKF can be replaced by the sample covariance $\tilde{\Gamma}$ of the $\{\eta_{j+1}^{(n)}\}_{n=1}^{N}$ and this is often done in practice. The sample covariance of the updated ensemble in this case is equal to $(I - \tilde{K}_{j+1}H)\widehat{C}_{j+1}$ where $\tilde{K}_{j+1}$ is the gain corresponding to the sample covariance $\tilde{\Gamma}$.

There are a range of parameters which can be used to tune the approximate Gaussian filters or modifications of those filters. In practical implementations, especially for high dimensional problems, the basic forms of the ExKF and EnKF as described here are prone to poor behaviour and such tuning is essential [66, 40]. In Examples 4.12 and 4.13 we have already shown the role of variance inflation for 3DVAR and this type of approach



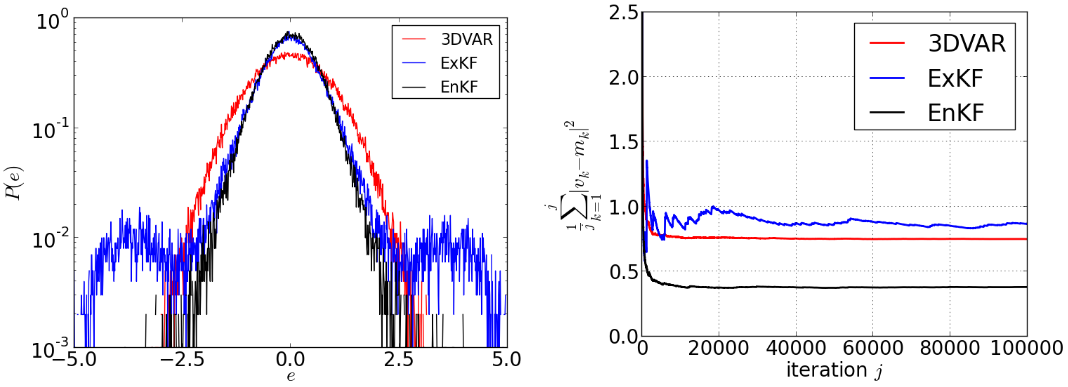
(a) Solution.

(b) Covariance.

(c) Error.

Figure 4.10: Particle Filter (optimal proposal) on the sin map Example 2.3 with $\alpha = 2.5$, $\sigma = 0.3$, $\gamma = 1$ and $N = 100$, see also `p15.m` in section 5.3.8.

is also fruitfully used within ExKF and EnKF. A basic version of variance inflation is to replace the estimate $\widehat{C}_{j+1}$ in (4.13) by $\epsilon\widehat{C} + \widehat{C}_{j+1}$ where $\widehat{C}$ is a fixed covariance such as that used in a 3DVAR method. Introducing $\epsilon \in (0,1)$ leads, for positive-definite $\widehat{C}$, to an operator without a null-space and consequently to better behaviour. In contrast taking $\epsilon = 0$ can lead to singular model covariances. This observation is particularly important when the EnKF is used in high dimensional systems where the number of ensemble members, N, is always less than the dimension n of the state space. In this situation $\widehat{C}_{j+1}$ necessarily has a null-space of dimension at least $n - N$. It can also be important for the ExKF where the evolving dynamics can lead, asymptotically in j, to degenerate $\widehat{C}_{j+1}$ with non-trivial null-space. Notice also that this form of variance inflation can be thought of as using 3DVAR-like covariance updates, in the directions not described by the ensemble covariance. This can be beneficial in terms of the ideas underlying Theorem 4.10 where the key idea is that K close to the identity can help ameliorate growth in the underlying dynamics. This may also be achieved by replacing the estimate $\widehat{C}_{j+1}$ in (4.13) by $(1+\epsilon)\widehat{C}_{j+1}$. This is another commonly used inflation tactic; note, however, that it lacks the benefit of rank correction. It may therefore be combined with the additive inflation yielding $\epsilon_1\widehat{C} + (1+\epsilon_2)\widehat{C}_{j+1}$. More details regarding tuning of filters through inflation can be found in [4, 42, 63, 66, 40].



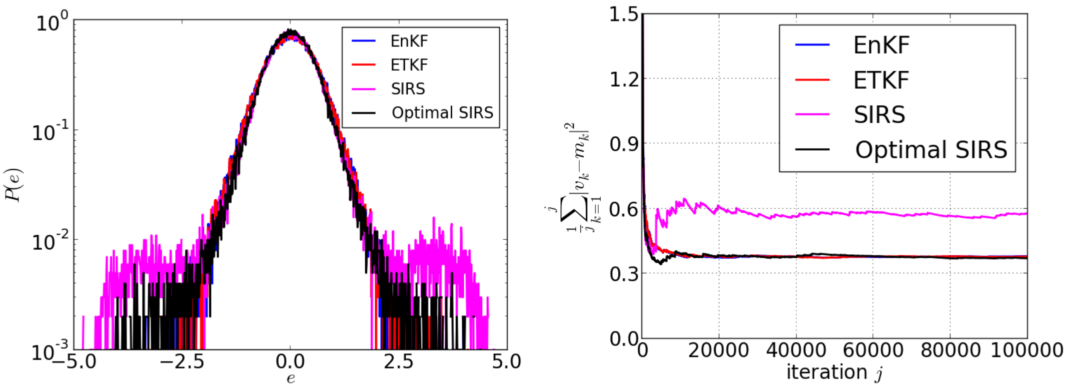
(a) Log-scale histograms.

(b) Running average root mean square e.

Figure 4.11: Convergence of $e = m - v^{\dagger}$ for each filter for the sin map Example 2.3, corresponding to solutions from Figs. 4.5, 4.6, 4.7.

Another methodology which is important for practical implementation of the EnKF is **localization** [66, 40]. This is used to reduce unwanted correlations in $\widehat{C}_j$ between points which are separated by large distances in space. The underlying assumption is that the correlation between points decays proportionally to their distance from one another, and as such is increasingly corrupted by the sample error in ensemble methods. The sample covariance is hence modified to remove correlations between points separated by large distances in space. This is typically achieved by composing the empirical correlation matrix with a convolution kernel. Localization can have the further benefit of increasing rank, as for the first type of variance inflation described above. An early reference illustrating the benefits and possible implementation of localization is [58]. An important reference which links this concept firmly with ideas from dynamical systems is [94].

Following the great success of the ensemble Kalman filter algorithm, in a series of papers [110, 18, 3, 118], the square-root filter framework was (re)discovered. The idea goes back to at least [5]. We focused the discussion above in section 4.2.4 on the ETKF, but we note that it is possible to derive different transformations. For example, the singular evolutive interpolated Kalman (SEIK) filter proceeds by first projecting the ensemble into the $(K-1)$-dimensional mean-free subspace, and then identifying a $(K-1)\times(K-1)$ matrix transformation, effectively prescribing a $K \times (K-1)$ matrix transformation L_j as opposed to the $K \times K$ rank $(K-1)$ matrix $T_j^{1/2}$ proposed in ETKF. The former is unique up to unitary transformation, while the latter is unique only up to unitary transformations which have $\mathbf{1}$ as eigenvector. Other alternative transformations may take the forms A_j or $\tilde{K}_j$ such that $X_j = A_j\widehat{X}_j$ or $X_j = (I - \tilde{K}H)\widehat{X}_j$. These are known as the ensemble adjustment Kalman filter (EAKF) and the ensemble square-root filter (ESRF) respectively. See [18] for details about the ETKF, [3] for details about the EAKF and [118] for details about the ESRF [118]. A review of all three is given in [110]. The similar singular evolutive interpolated Kalman (SEIK) filter was introduced in [96] and is compared with the other square root filters in [89]. Other ensemble-based filters have been developed in recent years bridging the ensemble Kalman filter with the particle filter, for example [57, 56, 98, 102].



(a) Log-scale histograms.

(b) Running average root mean square e.

Figure 4.12: Convergence of $e = m - v^\dagger$ for both versions of EnKF in comparison to the particle filters for the sin map Ex. 1.3, corresponding to solutions from Figs. 4.7, 4.8, 4.9, and 4.10.

- Section 4.3. In the linear case, the extended Kalman filter of course coincides with the Kalman filter; furthermore, in this case the perturbed observation ensemble Kalman filter reproduces the true posterior distribution in the large particle limit [40]. However the filters introduced in section 4.2 do not produce the correct posterior distribution when applied to general nonlinear problems. On the contrary, the particle filter *does* recover the true posterior distribution as the number of particles tends to infinity, as we show in Theorem 4.5. This proof is adapted from the very clear exposition in [97].

  For more refined analyses of the convergence of particle filters see, for example, [33, 37] and references therein. As explained in Remarks 4.6 the constant appearing in the convergence results may depend exponentially on time if the mixing properties of the transition kernel $\mathbb{P}(dv_j|v_{j-1})$ are poor (the undesirable properties of deterministic dynamics illustrate this). There is also interesting work studying the effect of the dimension [103]. A proof which exploits ergodicity of the transition kernel, when that is present, may be found in [37]; the assumptions there on the transition and observation kernels are very strong, and are generally not satisfied in practice, but studies indicate that comparable results may hold under less stringent conditions.

  For a derivation and discussion of the optimal proposal , introduced in section 4.3.3, see [38] and references therein. We also mention here the implicit filters developed by Chorin and co-workers [27, 26, 25]. These involve solving an implicit nonlinear equation for each particle which includes knowledge of the next set of observed data. This has some similarities to the method proposed in [115] and both are related to the optimal proposal mentioned above.

- Section 4.4. The stability of the Kalman filter is a well-studied subject and the book [68] provides an excellent overview from the perspective of linear algebra. For extensions to the extended Kalman filter see [63]. Theorem 4.10 provides a glimpse into the mechanisms at play within 3DVAR, and approximate Gaussian filters in general, in determining stability and accuracy: the incorporation of data can convert unstable dynamical systems, with positive Lyapunov exponents, into contractive non-autonomous dynamical systems, thereby leading, in the case of small observational noise, to filters which recover the true signal within a small error. This idea was highlighted in [24] and first studied rigorously for the 3DVAR method applied to the Navier-Stokes equation in [21]; this work was subsequently generalized to a variety of different models in [88, 70, 69]. It is also of note that these analyses of 3DVAR build heavily on ideas developed in [54] for a specialized form of data assimilation in which the observations are noise-free. In the language of the synchronization filter introduced in subsection 4.4.3, this paper demonstrates the synchronization property (4.47) for the Navier-Stokes equation with sufficiently large number of Fourier mode observations, and for the Lorenz '63 model of Example 2.6 observed in only the first component. The paper [69] consider similar issues for the Lorenz '96 model of Example 2.7. Similar ideas are studied for the perturbed observation EnKF in [67]; in this case it is necessary to introduce a form a variance inflation to get a result analogous to Theorem 4.10. An important step in the theoretical analysis of ensemble Kalman filter methods is the paper [48] which uses ideas from the theory of shadowing in dynamical systems; the work proves that the ETKF variant can shadow the truth on arbitrarily long time intervals, provided the dimension of the ensemble is greater than the number of unstable directions in the system.

In the context of filter stability it is important to understand the *optimality* of the mean of the true filtering distribution . We observe that all of the filtering algorithms that we have described produce an estimate of the probability distribution $\mathbb{P}(v_j|Y_j)$ that depends only on the data Y_j. There is a precise sense in which the true filtering distribution can be used to find a lower bound on the accuracy that can be achieved by any of these approximate algorithms. We let $\mathbb{E}(v_j|Y_j)$ denote the mean of v_j under the probability distribution $\mathbb{P}(v_j|Y_j)$ and let $E_j(Y_j)$ denote any estimate of the state v_j based only on data Y_j. Now consider all possible random data sets Y_j generated by the model (2.1), (2.2), noting that the randomness is generated by the initial condition v_0 and the noises $\{\xi_j, \eta_j\}$; in particular, conditioning on Y_j to obtain the probability distribution $\mathbb{P}(v_j|Y_j)$ can be thought of as being induced by conditioning on the observational noise $\{\eta_k\}_{k=1,\dots,j}$. Then $E_j^*(Y_j) := \mathbb{E}(v_j|Y_j)$ minimizes the mean-square error with respect to the random model (2.1), (2.2) [80, 63, 65]:

$$\mathbb{E}\|v_j - E_j^*(Y_j)\|^2 \le \mathbb{E}\|v_j - E_j(Y_j)\|^2 \tag{4.49}$$

for all $E_j(Y_j)$. Thus the algorithms we have described can do no better at estimating the state of the system than can be achieved, in principle, from the conditional mean of the state given the data $\mathbb{E}(v_j|Y_j)$. This lower bound holds on average over all instances of the model. An alternative way to view the inequality (4.49) is as a means to providing upper bounds on the true filter. For example, under the conditions of Theorem 4.10 the righthand side of (4.49) is, asymptotically as $j \to \infty$, of size $\mathcal{O}(\epsilon^2)$; thus we deduce that

$$\limsup_{j\to\infty} \mathbb{E}\|v_j - \mathbb{E}(v_j|Y_j)\|^2 \le C\epsilon^2.$$

This viewpoint is adopted in [100] where the 3DVAR filter is used to bound the true filtering distribution. This latter optimality property can be viewed as resulting from the Galerkin orthogonality interpretation of the error resulting from taking conditional expectation.

We have considered large time-behaviour on the assumption that the map Ψ can be implemented exactly. In situations where the underlying map Ψ arises from a differential equation and numerical methods are required, large excursions in phase space caused by observational noise can cause numerical instabilities in the integration methods underlying the filters. Remark **??** illustrates this fact in the context of continuous time. See [49] for a discussion of this issue.

- Section 4.5 We mention here the *rank histogram*. This is another consistency check on the output of ensemble or particle based approximations of the filtering distribution . The idea is to consider scalar observed quantities consisting of generating ordered bins associated to that scalar and then keeping track of the statistics over time of the data y_j with respect to the bins. For example, if one has an approximation of the distribution consisting of N equally-weighted particles, then a rank histogram for the first component of the state consists of three steps, each carried out at each time j. First, add a random draw from the observational noise $N(0, \Gamma)$ to each particle after the prediction phase of the algorithm. Secondly order the particles according to the value of their first component, generating $N-1$ bins between the values of the first component of each particle, and with one extra bin on each end. Finally, rank the current observation y_j between 1 and $N+1$ depending on which bin it lands in. Proceeding to do this at each time j, a histogram of the rank of the observations is obtained. The "spread" of the ensemble can be evaluated using this diagnostic. If the histogram is uniform, then the spread is consistent. If it is concentrated to the center, then the spread is overestimated. If it is concentrated at the edges, then the spread is underestimated. This consistency check on the statistical model used was introduced in [2] and is widely adopted throughout the data assimilation community.

4.7 Exercises

1. Consider the Kalman filter in the case where $M = H = I$, $\Sigma = 0$ and $\Gamma > 0$. Prove that the covariance operator C_j converges to 0 as $j \to \infty$. Modify the program `p8.m` so that it applies to this set-up, in the two dimension setting. Verify what you have proved regarding the covariance and make a conjecture about the limiting behaviour of the mean of the Kalman filter.

2. Consider the 3DVAR algorithm in the case where $\Psi(v) = v, H = I$, $\Sigma = 0$ and $\Gamma > 0$. Choose $\widehat{C} = \alpha\Gamma$. Find an equation for the error $e_j := v_j - m_j$ and derive an expression for $\lim_{j\to\infty} \mathbb{E}|e_j|^2$ in terms of α and $\sigma^2 := \mathbb{E}|\eta_j|^2$. Modify the program `p9.m` so that it applies to this set-up, in the one dimensional setting. Verify what you have proved regarding the limiting behaviour of the m_j.

3. Consider the EnKF algorithm in the same setting as the previous example. Modify program `p12.m` so that it applies to this set-up, in the one dimensional setting. Study the behaviour of the sequence m_j found as the mean of the particles $v_j^{(n)}$ over the ensemble index n.

4. Consider the SIRS algorithm in the same setting as the previous example. Modify program `p14.m` so that it applies to this set-up, in the one dimensional setting. Study the behaviour of the sequence m_j found as the mean of the particles $v_j^{(n)}$ over the ensemble index n.

5. Make comparative comments regarding the 3DVAR, EnKF and SIRS algorithms, on the basis of your solutions to the three preceding exercises.

6. In this exercise we study the behaviour of the mean of the Kalman filter in the case of one dimensional dynamics. The notation follows the development in subsection 4.4.1.

Consider the case $\sigma = 0$ and assume that the data $\{y_j\}_{j\in\mathbb{N}}$ is generated from a true signal $\{v_j^\dagger\}_{j\in\mathbb{Z}^+}$ governed by the equation

$$v_{j+1}^\dagger = \lambda v_j^\dagger,$$

and that the additive observational noise $\{\eta_j\}_{j\in\mathbb{N}}$ is drawn from an i.i.d. sequence with variance γ^2. Define the error $e_j = m_j - v_j^\dagger$ between the estimated mean and the true signal and use (4.37b) to show that

$$e_{j+1} = \left(1 - \frac{c_{j+1}}{\gamma^2}\right)\lambda e_j + \frac{c_{j+1}}{\gamma^2}\eta_{j+1}. \tag{4.50}$$

Deduce that e_j is Gaussian and that its mean and covariance satisfy the equations

$$\mathbb{E}e_{j+1} = \lambda\left(1 - \frac{c_{j+1}}{\gamma^2}\right)\mathbb{E}e_j, \tag{4.51}$$

and

$$\mathbb{E}e_{j+1}^2 = \lambda^2\left(1 - \frac{c_{j+1}}{\gamma^2}\right)^2\mathbb{E}e_j^2 + \frac{c_{j+1}^2}{\gamma^2}. \tag{4.52}$$

Equation (4.51) can be solved to obtain

$$\mathbb{E}e_j = \lambda^j\left[\prod_{i=0}^{j}\left(1 - \frac{c_{i+1}}{\gamma^2}\right)\right]\mathbb{E}e_0, \tag{4.53}$$

and, in a similar way, obtain for the solution of (4.52):

$$\mathbb{E}e_j^2 = \lambda^{2j}\left[\prod_{i=0}^{j-1}\left(1 - \frac{c_{i+1}}{\gamma^2}\right)^2\right]\mathbb{E}e_0^2 + \sum_{i=0}^{j-1}\left\{\left[\prod_{k=i+1}^{j}\left(1 - \frac{c_k}{\gamma^2}\right)\right]\lambda^{2(j-i)}\frac{c_i^2}{\gamma^2}\right\} + \frac{c_j^2}{\gamma^2} \tag{4.54}$$

Using the properties of the variance derived in 4.4.1, prove that the mean of the error tends to zero and that the asymptotic variance is bounded by γ^2.

Chapter 5

Discrete Time: MATLAB Programs

This chapter is dedicated to illustrating the examples, theory and algorithms, as presented in the previous chapters, through a few short and easy to follow MATLAB programs. These programs are provided for two reasons: (i) for some readers they will form the best route by which to appreciate the details of the examples, theory and algorithms we describe; (ii) for other readers they will be a useful starting point to develop their own codes: whilst ours are not necessarily the optimal implementations of the algorithms discussed in these notes, they have been structured to be simple to understand, to modify and to extend. In particular the code may be readily extended to solve problems more complex than those described in the Examples 2.1-2.7 which we will use for most of our illustrations. The chapter is divided into three sections, corresponding to programs relevant to each of the preceding three chapters.

Before getting into details we highlight a few principles that have been adopted in the programs and in accompanying text of this chapter. First, notation is consistent between programs, and matches the text in the previous sections of the book as far as possible. Second, since many of the elements of the individual programs are repeated, they will be described in detail only in the text corresponding to the program in which they first appear; the short annotations explaining them will be repeated within the programs however. Third, the reader is advised to use the documentation available at the command line for *any* built-in functions of MATLAB; this information can be accessed using the `help` command – for example the documentation for the command `help` can be accessed by typing `help help`.

5.1 Chapter 2 Programs

The programs `p1.m` and `p2.m` used to generate the figures in Chapter 2 are presented in this section. Thus these algorithms simply solve the dynamical system (2.1), and process the resulting data.

5.1.1. p1.m

The first program `p1.m` illustrates how to obtain sample paths from equations (2.1) and (2.3). In particular the program simulates sample paths of the equation

$$u_{j+1} = \alpha \sin(u_j) + \xi_j, \tag{5.1}$$

with $\xi_j \sim N(0,\sigma^2)$ i.i.d. and $\alpha = 2.5$, both for deterministic ($\sigma = 0$) and stochastic dynamics ($\sigma \neq 0$) corresponding to Example 2.3. In line 5 the variable `J` is defined, which corresponds to the number of forward steps that we will take. The parameters α and σ are set in lines 6-7. The seed for the random number generator is set to `sd`$\in \mathbb{N}$ in line 8 using the command `rng(sd)`. This guarantees the results will be reproduced exactly by running the program with this same `sd`. Different choices of `sd`$\in \mathbb{N}$ will lead to different streams of random numbers used in the program, which may also be desirable in order to observe the effects of different random numbers on the output. The command `sd` will be called in the preamble of all of the programs that follow. In line 9, two vectors of length `J` are created named `v` and `vnoise`; after running the program, these two vectors contain the solution for the case of deterministic ($\sigma = 0$) and stochastic dynamics ($\sigma = 0.25$) respectively. After setting the initial conditions in line 10, the desired map is iterated, without and with noise, in lines $12 - 15$. Note that the only difference between the forward iterations of `v` and `vnoise` is the presence of the `sigma*randn` term, which corresponds to the generation of a random variable sampled from $N(0, \sigma^2)$. Lines 17-18 contain code which graphs the trajectories, with and without noise, to produce Figure 2.3. Figures 2.1,2.2 and 2.5 were obtained by simply modifying lines $12 - 15$ of this program, in order to create sample paths for the corresponding Ψ for the other three examples; furthermore, Figure 2.4a was generated from output of this program and Figure 2.4b was generated from output of a modification of this program.

```
 1  clear;set(0,’defaultaxesfontsize’,20);format long
 2  %%% p1.m - behaviour of sin map (Ex. 1.3)
 3  %%% with and without observational noise
 4
 5  J=10000;% number of steps
 6  alpha=2.5;% dynamics determined by alpha
 7  sigma=0.25;% dynamics noise variance is sigma^2
 8  sd=1;rng(sd);% choose random number seed
 9  v=zeros(J+1,1);  vnoise=zeros(J+1,1);% preallocate space
10  v(1)=1;vnoise(1)=1;% initial conditions
11
12  for i=1:J
13      v(i+1)=alpha*sin(v(i));
14      vnoise(i+1)=alpha*sin(vnoise(i))+sigma*randn;
15  end
16
17  figure(1), plot([0:1:J],v),
18  figure(2), plot([0:1:J],vnoise),
```

5.1.2. p2.m

The second program presented here, `p2.m`, is designed to visualize the posterior distribution in the case of one dimensional deterministic dynamics . For clarity, the program is separated into three main sections. The `setup` section in lines 5-10 defines the parameters of the problem. The model parameter `r` is defined in line 6, and determines the dynamics of the forward model, in this case given by the logistic map (2.9):

$$v_{j+1} = rv_j(1 - v_j). \tag{5.2}$$

The dynamics are taken as deterministic, so the parameter `sigma` does not feature here. The parameter `r`$= 2$ so that the dynamics are not chaotic, as the explicit solution given in Example 2.4 shows. The parameters `m0` and `C0` define the mean and covariance of the prior distribution $v_0 \sim N(m_0, C_0)$, whilst `gamma` defines the observational noise $\eta_j \sim N(0, \gamma^2)$.

The `truth` section in lines 14-20 generates the true reference trajectory (or, truth) `vt` in line 18 given by (5.2), as well as the observations `y` in line 19 given by

$$y_j = v_j + \eta_j. \tag{5.3}$$

Note that the index of `y(:,j)` corresponds to observation of `H*v(:,j+1)`. This is due to the fact that the first index of an array in matlab is `j=1`, while the initial condition is v_0, and the first observation is of v_1. So, effectively the indices of `y` are correct as corresponding to the text and equation (5.3), but the indices of `v` are one off. The memory for these vectors is *preallocated* in line 14. This is not necessary because MATLAB would simply *dynamically allocate* the memory in its absence, but it would slow down the computations due to the necessity of allocating new memory each time the given array changes size. Commenting this line out allows observation of this effect, which becomes significant when `J` becomes sufficiently large.

The `solution` section after line 24 computes the solution, in this case the point-wise representation of the posterior smoothing distribution on the scalar initial condition. The point-wise values of initial condition are given by the vector `v0` (v_0) defined in line 24. There are many ways to construct such vectors, this convention defines the initial (0.01) and final (0.99) values and a uniform step size 0.0005. It is also possible to use the command `v0=linspace(0.01,0.99,1961)`, defining the *number* 1961 of intermediate points, rather than the stepsize 0.0005. The corresponding vector of values of `Phidet` ($\Phi_{\rm det}$), `Jdet` ($\mathsf{J}_{\rm det}$), and `Idet` ($\mathsf{I}_{\rm det}$) are computed in lines 32, 29, and 34 for each value of `v0`, as related by the equation

$$\mathsf{I}_{\rm det}(v_0; y) = \mathsf{J}_{\rm det}(v_0) + \Phi_{\rm det}(v_0; y), \tag{5.4}$$

where $\mathsf{J}_{\rm det}(v_0)$ is the **background** penalization and $\Phi_{\rm det}(v_0; y)$ is the **model-data misfit** functional given by (2.29b) and (2.29c) respectively. The function $\mathsf{I}_{\rm det}(v_0; y)$ is the negative log-posterior as given in Theorem 2.11. Having obtained $\mathsf{I}_{\rm det}(v_0; y)$ we calculate $\mathbb{P}(v_0|y)$ in lines 37-38, using the formula

$$\mathbb{P}(v_0|y) = \frac{\exp(-\mathsf{I}_{\rm det}(v_0; y))}{\int \exp(-\mathsf{I}_{\rm det}(v_0; y))}. \tag{5.5}$$

The trajectory v corresponding to the given value of v_0 (`v0(i)`) is denoted by `vv` and is replaced for each new value of `v0(i)` in lines 29 and 31 since it is only required to compute `Idet`. The command `trapz(v0,exp(-Idet))` in line 37 approximates the denominator of the above by the trapezoidal rule, i.e. the summation

$$\mathtt{trapz(v0, exp(-Idet))} = \sum_{i=1}^{N-1} \mathtt{(v0(i+1) - v0(i)) * (Idet(i+1) + Idet(i))/2}. \tag{5.6}$$

The rest of the program deals with plotting our results and in this instance it coincides with the output of Figure 2.11b. Again simple modifications of this program were used to produce Figures 2.10, 2.12 and 2.13. Note that `rng(sd)` in line 8 allows us to use the same random numbers every time the file is executed; those random numbers are generated with the seed `sd` as described in the previous section 5.1.1. Commenting this line out would result in the creation of new realizations of the random data y, different from the ones used to obtain Figure 2.11b.

```
 1  clear; set(0,’defaultaxesfontsize’,20); format long
 2  %%% p2.m smoothing problem for the deterministic logistic map (Ex. 1.4)
 3  %% setup
 4
 5  J=1000;% number of steps
 6  r=2;% dynamics determined by r
 7  gamma=0.1;% observational noise variance is gammaˆ2
 8  C0=0.01;% prior initial condition variance
 9  m0=0.7;% prior initial condition mean
10  sd=1;rng(sd);% choose random number seed
11
12  %% truth
13
14  vt=zeros(J+1,1); y=zeros(J,1);% preallocate space to save time
15  vt(1)=0.1;% truth initial condition
16  for j=1:J
17      % can be replaced by Psi for each problem
18      vt(j+1)=r*vt(j)*(1-vt(j));% create truth
19      y(j)=vt(j+1)+gamma*randn;% create data
20  end
21
22  %% solution
23
24  v0=[0.01:0.0005:0.99];% construct vector of different initial data
25  Phidet=zeros(length(v0),1);Idet=Phidet;Jdet=Phidet;% preallocate space
26  vv=zeros(J,1);% preallocate space
27  % loop through initial conditions vv0, and compute log posterior I0(vv0)
28  for j=1:length(v0)
29     vv(1)=v0(j); Jdet(j)=1/2/C0*(v0(j)-m0)ˆ2;% background penalization
30       for i=1:J
31          vv(i+1)=r*vv(i)*(1-vv(i));
32          Phidet(j)=Phidet(j)+1/2/gammaˆ2*(y(i)-vv(i+1))ˆ2;% misfit
33       end
34      Idet(j)=Phidet(j)+Jdet(j);
35  end
36
37  constant=trapz(v0,exp(-Idet));% approximate normalizing constant
38  P=exp(-Idet)/constant;% normalize posterior distribution
39  prior=normpdf(v0,m0,sqrt(C0));% calculate prior distribution
40
41  figure(1),plot(v0,prior,’k’,’LineWidth’,2)
42  hold on, plot(v0,P,’r--’,’LineWidth’,2), xlabel ’v_0’,
43  legend ’prior’ J=10ˆ3
```

5.2 Chapter 3 Programs

The programs p3.m-p7.m, used to generate the figures in Chapter 3, are presented in this section. Hence various MCMC algorithms used to sample the posterior smoothing distribution are given. Furthermore, optimization algorithms used to obtain solution of the 4DVAR and w4DVAR variational methods are also introduced. Our general theoretical development of MCMC methods in section 3.2 employs a notation of u for the state of the chain and w for the proposal. For deterministic dynamics the state is the initial condition v_0; for stochastic dynamics it is either the signal v or the pair (v_0, ξ) where ξ is the noise (since this pair determines the signal). Where appropriate, the programs described here use the letter v, and variants on it, for the state of the Markov chain, to keep the connection with the underlying dynamics model.

5.2.1. p3.m

The program p3.m contains an implementation of the Random Walk Metropolis (RWM) MCMC algorithm. The development follows section 3.2.3 where the algorithm is used to determine the posterior distribution on the initial condition arising from the deterministic logistic map of Example 2.4 given by (5.2). Note that in this case, since the the underlying dynamics are deterministic and hence completely determined by the initial condition, the RWM algorithm will provide samples from a probability distribution on $\mathbb{R}$.

As in program p2.m, the code is divided into 3 sections: setup where parameters are defined, truth where the truth and data are generated, and solution where the solution is computed, this time by means of MCMC samples from the posterior smoothing distribution. The parameters in lines 5-10 and the true solution (here taken as only the initial condition, rather than the trajectory it gives rise to) vt in line 14 are taken to be the same as those used to generate Figure 2.13. The temporary vector vv generated in line 19 is the trajectory corresponding to the truth (vv(1)=vt in line 14), and used to calculate the observations y in line 20. The true value vt will also be used as the initial sample in the Markov chain for this and for all subsequent MCMC programs. This scenario is, of course, not possible in the case that the data is not simulated. However it is useful in the case that the data is simulated, as it is here, because it can reduce the burn-in time, i.e. the time necessary for the current sample in the chain to reach the target distribution, or the high-probability region of the state-space. Because we initialize the Markov chain at the truth, the value of $\mathsf{I}_{\mathrm{det}}(v^\dagger)$, denoted by the temporary variable Idet, is required to determine the initial acceptance probability, as described below. It is computed in lines 15-23 exactly as in lines 25-34 of program p2.m, as described around equation (5.4).

In the solution section some additional MCMC parameters are defined. In line 28 the number of samples is set to N $=10^5$. For the parameters and specific data used here, this is sufficient for convergence of the Markov chain. In line 30 the step-size parameter beta is pre-set such that the algorithm for this particular posterior distribution has a reasonable acceptance probability, or ratio of accepted vs. rejected moves. A general rule of thumb for this is that it should be somewhere around 0.5, to ensure that the algorithm is not too correlated because of high rejection rate (acceptance probability near zero) and that it is not too correlated because of small moves (acceptance probability near one). The vector V defined in line 29 will save all of the samples. This is an example where preallocation is *very* important. Try using the commands tic and toc before and respectively after the loop in lines 33-50 in order to time the chain both with and without preallocation. [1] In line 34 a

[1] In practice, one may often choose to collect certain statistics from the chain "on-the-fly" rather than

move is proposed according to the proposal equation (3.15):

$$w^{(k)} = v^{(k-1)} + \beta\iota^{(k-1)}$$

where v(v) is the current state of the chain (initially taken to be equal to the true initial condition v_0), $\iota^{(k-1)}$=randn is an i.i.d. standard normal, and w represents $w^{(k)}$. Indices are not used for v and w because they will be overwritten at each iteration.

The temporary variable vv is again used for the trajectory corresponding to $w^{(k)}$ as a vehicle to compute the value of the proposed $\mathsf{I}_{\text{det}}(w^{(k)}; y)$, denoted in line 42 by I0prop = J0prop + Phiprop. In lines 44-46 the decision to accept or reject the proposal is made based on the acceptance probability

$$a(v^{(k-1)}, w^{(k)}) = 1 \wedge \exp(\mathsf{I}_{\text{det}}(v^{(k-1)}; y) - \mathsf{I}_{\text{det}}(w^{(k)}; y)).$$

In practice this corresponds to drawing a uniform random number rand and replacing v and Idet in line 45 with w and I0prop if rand<exp(I0-I0prop) in line 44. The variable bb is incremented if the proposal is accepted, so that the running ratio of accepted moves bb to total steps n can be computed in line 47. This approximates the average acceptance probability. The current sample $v^{(k)}$ is stored in line 48. Notice that here one could replace v by V(n-1) in line 34, and by V(n) in line 45, thereby eliminating v and line 48, and letting w be the only temporary variable. However, the present construction is favourable because, as mentioned above, in general one may not wish to save every sample.

The samples V are used in lines 51-53 in order to visualize the posterior distribution. In particular, bins of width dx are defined in line 51, and the command hist is used in line 52. The assignment Z = hist(V,v0) means first the real-number line is split into M bins with centers defined according to v0(i) for $i = 1, \dots, M$, with the first and last bin corresponding to the negative, respectively positive, half-lines. Second, Z(i) counts the number of k for which V(k) is in the bin with center determined by v0(i). Again, trapz (5.6) is used to compute the normalizing constant in line 53, directly within the plotting command. The choice of the location of the histogram bins allows for a direct comparison with the posterior distribution calculated from the program p2.m by directly evaluating $\mathsf{I}_{\text{det}}(v; y)$ defined in (5.4) for different values of initial conditions v. This output is then compared with the corresponding output of p2.m for the same parameters in Figure 3.2.

saving every sample, particularly if the state-space is high-dimensional where the memory required for each sample is large.

```
clear; set(0,’defaultaxesfontsize’,20); format long
%%% p3.m MCMC RWM algorithm for logistic map (Ex. 1.4)
%% setup

J=5;% number of steps
r=4;% dynamics determined by alpha
gamma=0.2;% observational noise variance is gamma^2
C0=0.01;% prior initial condition variance
m0=0.5;% prior initial condition mean
sd=10;rng(sd);% choose random number seed

%% truth

vt=0.3;vv(1)=vt;% truth initial condition
Jdet=1/2/C0*(vt-m0)^2;% background penalization
Phidet=0;% initialization model-data misfit functional
for j=1:J
    % can be replaced by Psi for each problem
    vv(j+1)=r*vv(j)*(1-vv(j));% create truth
    y(j)=vv(j+1)+gamma*randn;% create data
    Phidet=Phidet+1/2/gamma^2*(y(j)-vv(j+1))^2;% misfit functional
end
Idet=Jdet+Phidet;% compute log posterior of the truth

%% solution
% Markov Chain Monte Carlo: N forward steps of the
% Markov Chain on R (with truth initial condition)
N=1e5;% number of samples
V=zeros(N,1);% preallocate space to save time
beta=0.05;% step-size of random walker
v=vt;% truth initial condition (or else update I0)
n=1; bb=0; rat(1)=0;
while n<=N
    w=v+sqrt(2*beta)*randn;% propose sample from random walker
    vv(1)=w;
    Jdetprop=1/2/C0*(w-m0)^2;% background penalization
    Phidetprop=0;
    for i=1:J
            vv(i+1)=r*vv(i)*(1-vv(i));
            Phidetprop=Phidetprop+1/2/gamma^2*(y(i)-vv(i+1))^2;
    end
    Idetprop=Jdetprop+Phidetprop;% compute log posterior of the proposal

    if rand<exp(Idet-Idetprop)% accept or reject proposed sample
        v=w; Idet=Idetprop; bb=bb+1;% update the Markov chain
    end
    rat(n)=bb/n;% running rate of acceptance
    V(n)=v;% store the chain
    n=n+1;
end
dx=0.0005; v0=[0.01:dx:0.99];
Z=hist(V,v0);% construct the posterior histogram
figure(1), plot(v0,Z/trapz(v0,Z),’k’,’Linewidth’,2)% visualize posterior
```

5.2.2. p4.m

The program `p4.m` contains an implementation of the Independence Dynamics Sampler for stochastic dynamics , as introduced in subsection 3.2.4. Thus the posterior distribution is on the entire signal $\{v_j\}_{j\in\mathbb{J}}$. The forward model in this case is from Example 2.3, given by (5.1). The smoothing distribution $\mathbb{P}(v|Y)$ is therefore over the state-space $\mathbb{R}^{J+1}$.

The sections `setup`, `truth`, and `solution` are defined as for program `p3.m`, but note that now the smoothing distribution is over the entire path, not just over the initial condition, because we are considering stochastic dynamics . Since the state-space is now the path-space, rather than the initial condition as it was in program `p3.m`, the truth `vt`$\in \mathbb{R}^{J+1}$ is now a vector. Its initial condition is taken as a draw from $N(m_0, C_0)$ in line 16, and the trajectory is computed in line 20, so that at the end `vt`$\sim \rho_0$. As in program `p3.m`, $v^\dagger$ (`vt`) will be the chosen initial condition in the Markov chain (to ameliorate burn-in issues) and so $\Phi(v^\dagger; y)$ is computed in line 23. Recall from subsection 3.2.4 that only $\Phi(\cdot; y)$ is required to compute the acceptance probability in this algorithm.

Notice that the collection of samples `V`$\in \mathbb{R}^{N\times J+1}$ preallocated in line 30 is substantial in this case, illustrating the memory issue which arises when the dimension of the signal space, and number of samples, increase.

The current state of the chain $v^{(k)}$, and the value of $\Phi(v^{(k)}; y)$ are again denoted `v` and `Phi`, while the proposal $w^{(k)}$ and the value of $\Phi(w^{(k)}; y)$ are again denoted `w` and `Phiprop`, as in program `p3`. As discussed in section 3.2.4, the proposal $w^{(k)}$ is an independent sample from the prior distribution ρ_0, similarly to $v^\dagger$, and it is constructed in lines 34-39. The acceptance probability used in line 40 is now

$$a(v^{(k-1)}, w^{(k)}) = 1 \wedge \exp(\Phi(v^{(k-1)}; y) - \Phi(w^{(k)}; y)), \tag{5.7}$$

The remainder of the program is structurally the same as `p3.m`. The outputs of this program are used to plot Figures 3.3, 3.4, and 3.5. Note that in the case of Figure 3.5, we have used $N = 10^8$ samples.

```
clear; set(0,'defaultaxesfontsize',20); format long
%%% p4.m MCMC INDEPENDENCE DYNAMICS SAMPLER algorithm
%%% for sin map (Ex. 1.3) with noise
%% setup

J=10;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=1;% dynamics noise variance is sigma^2
C0=1;% prior initial condition variance
m0=0;% prior initial condition mean
sd=0;rng(sd);% choose random number seed

%% truth

vt(1)=m0+sqrt(C0)*randn;% truth initial condition
Phi=0;

for j=1:J
    vt(j+1)=alpha*sin(vt(j))+sigma*randn;% create truth
    y(j)=vt(j+1)+gamma*randn;% create data
    % calculate log likelihood of truth, Phi(v;y) from (1.11)
    Phi=Phi+1/2/gamma^2*(y(j)-vt(j+1))^2;
end

%% solution
% Markov Chain Monte Carlo: N forward steps of the
% Markov Chain on R^{J+1} with truth initial condition
N=1e5;% number of samples
V=zeros(N,J+1);% preallocate space to save time
v=vt;% truth initial condition (or else update Phi)
n=1; bb=0; rat(1)=0;
while n<=N
    w(1)=sqrt(C0)*randn;% propose sample from the  prior
    Phiprop=0;
    for j=1:J
    w(j+1)=alpha*sin(w(j))+sigma*randn;% propose sample from the prior
    Phiprop=Phiprop+1/2/gamma^2*(y(j)-w(j+1))^2;% compute likelihood
    end
    if rand<exp(Phi-Phiprop)% accept or reject proposed sample
       v=w; Phi=Phiprop; bb=bb+1;% update the Markov chain
    end
   rat(n)=bb/n;% running rate of acceptance
   V(n,:)=v;% store the chain
   n=n+1;
end
% plot acceptance ratio and cumulative sample mean
figure;plot(rat)
figure;plot(cumsum(V(1:N,1))./[1:N]')
xlabel('samples N')
ylabel('(1/N) \Sigma_{n=1}^N v_0^{(n)}')
```

5.2.3. p5.m

The Independence Dynamics Sampler of subsection 5.2.2 may be very inefficient as typical random draws from the dynamics may be unlikely to fit the data as well as the current state, and will then be rejected. The fifth program `p5.m` gives an implementation of the pCN algorithm from section 3.2.4 which is designed to overcome this issue by including the parameter β which, if chosen small, allows for incremental steps in signal space and hence the possibility of non-negligible acceptance probabilities. This program is used to generate Figure 3.6

This program is almost identical to `p4.m`, and so only the points at which it differs will be described. First, since the acceptance probability is given by

$$a(v^{(k-1)}, w^{(k)}) = 1 \wedge \exp(\Phi(v^{(k-1)}; y) - \Phi(w^{(k)}; y) + G(v^{(k-1)}) - G(w^{(k)})),$$

the quantity

$$G(u) = \sum_{j=0}^{J-1} \Big(\frac{1}{2} |\Sigma^{-\frac{1}{2}} \Psi(u_j)|^2 - \langle \Sigma^{-\frac{1}{2}} u_{j+1}, \Sigma^{-\frac{1}{2}} \Psi(u_j) \rangle \Big)$$

will need to be computed, both for $v^{(k)}$ (denoted by `v` in lines 31 and 44) where its value is denoted by `G` ($v^{(0)} = v^\dagger$), as well as for $G(v^\dagger)$ is computed in line 22), and for $w^{(k)}$ (denoted by `w` in line 36) where its value is denoted by `Gprop` in line 39.

As discussed in section 3.2.4 the proposal $w^{(k)}$ is given by (3.19):

$$w^{(k)} = m + (1 - \beta^2)^{\frac{1}{2}} (v^{(k-1)} - m) + \beta \iota^{(k-1)}; \tag{5.8}$$

here $\iota^{(k-1)} \sim N(0, C)$ are i.i.d. and denoted by `iota` in line 35. C is the covariance of the Gaussian measure π_0 given in Equation (2.24) corresponding to the case of trivial dynamics $\Psi = 0$, and m is the mean of π_0. The value of m is given by `m` in line 33.

```
clear;set(0,'defaultaxesfontsize',20);format long
%%% p5.m MCMC pCN algorithm for sin map (Ex. 1.3) with noise

%% setup
J=10;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=.1;% dynamics noise variance is sigma^2
C0=1;% prior initial condition variance
m0=0;% prior initial condition mean
sd=0;rng(sd);% Choose random number seed

%% truth
vt(1)=m0+sqrt(C0)*randn;% truth initial condition
G=0;Phi=0;

for j=1:J
    vt(j+1)=alpha*sin(vt(j))+sigma*randn;% create truth
    y(j)=vt(j+1)+gamma*randn;% create data
    % calculate log density from (1.--)
    G=G+1/2/sigma^2*((alpha*sin(vt(j)))^2-2*vt(j+1)*alpha*sin(vt(j)));
    % calculate log likelihood phi(u;y) from (1.11)
    Phi=Phi+1/2/gamma^2*(y(j)-vt(j+1))^2;
end

%% solution
% Markov Chain Monte Carlo: N forward steps
N=1e5;% number of samples
beta=0.02;% step-size of pCN walker
v=vt;% truth initial condition (or update G + Phi)
V=zeros(N,J+1); n=1; bb=0; rat=0;
m=[m0,zeros(1,J)];
while n<=N
    iota=[sqrt(C0)*randn,sigma*randn(1,J)];% Gaussian prior sample
    w=m+sqrt(1-beta^2)*(v-m)+beta*iota;% propose pCN sample
    Gprop=0;Phiprop=0;
    for j=1:J
        Gprop=...
         Gprop+1/2/sigma^2*((alpha*sin(w(j)))^2-2*w(j+1)*alpha*sin(w(j)));
        Phiprop=Phiprop+1/2/gamma^2*(y(j)-w(j+1))^2;
    end
    if rand<exp(Phi-Phiprop+G-Gprop)% accept or reject proposed sample
        v=w;Phi=Phiprop;G=Gprop;bb=bb+1;% update the Markov chain
    end
    rat(n)=bb/n;% running rate of acceptance
    V(n,:)=v;% store the chain
    n=n+1;
end
% plot acceptance ratio and cumulative sample mean
figure;plot(rat)
figure;plot(cumsum(V(1:N,1))./[1:N]')
xlabel('samples N')
ylabel('(1/N) \Sigma_{n=1}^N v_0^{(n)}')
```

5.2.4. p6.m

The pCN dynamics sampler is now introduced as program `p6.m`. The Independence Dynamics Sampler of subsection 5.2.2 may be viewed as a special case of this algorithm for proposal variance $\beta = 1$. This proposal combines the benefits of tuning the step size β, while still respecting the prior distribution on the dynamics. It does so by sampling the initial condition and noise (v_0, ξ) rather than the path itself, in lines 34 and 35, as given by Equation (5.8). However, as opposed to the pCN sampler of the previous section, this variable `w` is now interpreted as a sample of (v_0, ξ) and is therefore fed into the path `vv` itself in line 39. The acceptance probability is the same as the Independence Dynamics Sampler (5.7), depending only on `Phi`. If the proposal is accepted, both the forcing `u=w` and the path `v=vv` are updated in line 44. Only the path is saved as in the previous routines, in line 47.

```
clear;set(0,'defaultaxesfontsize',20);format long
%%% p6.m MCMC pCN Dynamics algorithm for
%%% sin map (Ex. 1.3) with noise
%% setup

J=10;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=1;% dynamics noise variance is sigma^2
C0=1;% prior initial condition variance
m0=0;% prior initial condition mean
sd=0;rng(sd);% Choose random number seed

%% truth
vt(1)=m0+sqrt(C0)*randn;% truth initial condition
ut(1)=vt(1);
Phi=0;
for j=1:J
    ut(j+1)=sigma*randn;
    vt(j+1)=alpha*sin(vt(j))+ut(j+1);% create truth
    y(j)=vt(j+1)+gamma*randn;% create data
    % calculate log likelihood phi(u;y) from (1.11)
    Phi=Phi+1/2/gamma^2*(y(j)-vt(j+1))^2;
end

%% solution
% Markov Chain Monte Carlo: N forward steps
N=1e5;% number of samples
beta=0.2;% step-size of pCN walker
u=ut;v=vt;% truth initial condition (or update Phi)
V=zeros(N,J+1); n=1; bb=0; rat=0;m=[m0,zeros(1,J)];
while n<=N
    iota=[sqrt(C0)*randn,sigma*randn(1,J)];% Gaussian prior sample
    w=m+sqrt(1-beta^2)*(u-m)+beta*iota;% propose pCN sample
    vv(1)=w(1);
    Phiprop=0;
    for j=1:J
        vv(j+1)=alpha*sin(vv(j))+w(j+1);% create path
        Phiprop=Phiprop+1/2/gamma^2*(y(j)-vv(j+1))^2;
    end

    if rand<exp(Phi-Phiprop)% accept or reject proposed sample
        u=w;v=vv;Phi=Phiprop;bb=bb+1;% update the Markov chain
    end
       rat(n)=bb/n;% running rate of acceptance
       V(n,:)=v;% store the chain
       n=n+1;
end
% plot acceptance ratio and cumulative sample mean
figure;plot(rat)
figure;plot(cumsum(V(1:N,1))./[1:N]')
xlabel('samples N')
ylabel('(1/N) \Sigma_{n=1}^N v_0^{(n)}')
```

5.2.5. p7.m

The next program `p7.m` contains an implementation of the weak constrained variational algorithm w4DVAR discussed in section 3.3. This program is written as a function, whilst all previous programs were written as scripts. This choice was made for `p7.m` so that the MATLAB built-in function `fminsearch` can be used for optimization in the `solution` section, and the program can still be self-contained. To use this built-in function it is necessary to define an *auxiliary* objective function `I` to be optimized. The function `fminsearch` can be used within a script, but the auxiliary function would then have to be written separately, so we cannot avoid functions altogether unless we write the optimization algorithm by hand. We avoid the latter in order not to divert the focus of this text from the data assimilation problem, and algorithms to solve it, to the problem of how to optimize an objective function.

Again the forward model is that given by Example 2.8, namely (5.1). The `setup` and `truth` sections are similar to the previous programs, except that G, for example, need not be computed here. The auxiliary objective function `I` in this case is $\mathsf{I}(\cdot;y)$ from equation (2.21) given by

$$\mathsf{I}(\cdot;y) = \mathsf{J}(\cdot) + \Phi(\cdot;y), \tag{5.9}$$

where

$$\mathsf{J}(u) := \frac{1}{2}\left|C_0^{-\frac{1}{2}}(u_0 - m_0)\right|^2 + \sum_{j=0}^{J-1}\frac{1}{2}\left|\Sigma^{-\frac{1}{2}}\left(u_{j+1} - \Psi(u_j)\right)\right|^2, \tag{5.10}$$

and

$$\Phi(u;y) = \sum_{j=0}^{J-1}\frac{1}{2}\left|\Gamma^{-\frac{1}{2}}\left(y_{j+1} - h(u_{j+1})\right)\right|^2. \tag{5.11}$$

It is defined in lines 38-45. The auxiliary objective function takes as inputs (`u,y,sigma,gamma, alpha,m0, C0,J`), and gives output `out=` $\mathsf{I}(u;y)$ where $u \in R^{J+1}$ (given all the other parameters in its definition – the issue of identifying the input to be optimized over is discussed also below).

The initial guess for the optimization algorithm `uu` is taken as a standard normal random vector over $\mathbb{R}^{J+1}$ in line 27. In line 24, a standard normal random matrix of size 100^2 is drawn and thrown away. This is so one can easily change the input, e.g. to `randn(z)` for `z`$\in \mathbb{N}$, and induce different random initial vectors `uu` for the optimization algorithm, while keeping the data fixed by the random number seed `sd` set in line 12. The truth `vt` may be used as initial guess by uncommenting line 28. In particular, if the output of the minimization procedure is different for different initial conditions, then it is possible that the objective function $\mathsf{I}(\cdot;y)$ has multiple minima, and hence the posterior distribution $\mathbb{P}(\cdot|y)$ is multi-modal. As we have already seen in Figure 3.8 this is certainly true even in the case of scalar deterministic dynamics , when the underlying map gives rise to a chaotic flow.

The MATLAB optimization function `fminsearch` is called in line 32. The *function handle* command `@(u)I(u,` $\cdots$`)` is used to tell `fminsearch` that the objective function `I` is to be considered a function of `u`, even though it may take other parameter values as well (in this case, `y,sigma,gamma,alpha,m0,C0`, and `J`). The outputs of `fminsearch` are the value `vmap` such that `I(vmap)` is minimum, the value `fval = I(vmap)`, and the `exit flag` which takes the value 1 if the algorithm has converged. The reader is encouraged to use the `help` command for more details on this and other MATLAB functions used in the notes. The results of this minimization procedure are plotted in lines 34-35 together with the true value $v^\dagger$ as well as the data y. In Figure 3.9 such results are presented, including two minima which were found with different initial conditions.

```
function this=p7
clear;set(0,’defaultaxesfontsize’,20);format long
%%% p7.m weak 4DVAR for sin map (Ex. 1.3)
%% setup

J=5;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1e0;% observational noise variance is gamma^2
sigma=1;% dynamics noise variance is sigma^2
C0=1;% prior initial condition variance
m0=0;% prior initial condition mean
sd=1;rng(sd);% choose random number seed

%% truth

vt(1)=sqrt(C0)*randn;% truth initial condition
for j=1:J
    vt(j+1)=alpha*sin(vt(j))+sigma*randn;% create truth
    y(j)=vt(j+1)+gamma*randn;% create data
end

%% solution

    randn(100);% try uncommenting or changing the argument for different
               % initial conditions -- if the result is not the same,
               % there may be multimodality (e.g. 1 & 100).
    uu=randn(1,J+1);% initial guess
    %uu=vt;     % truth initial guess option

% solve with blackbox
% exitflag=1 ==> convergence
[vmap,fval,exitflag]=fminsearch(@(u)I(u,y,sigma,gamma,alpha,m0,C0,J),uu)

figure;plot([0:J],vmap,’Linewidth’,2);hold;plot([0:J],vt,’r’,’Linewidth’,2)
plot([1:J],y,’g’,’Linewidth’,2);hold;xlabel(’j’);legend(’MAP’,’truth’,’y’)

%% auxiliary objective function definition
function out=I(u,y,sigma,gamma,alpha,m0,C0,J)

Phi=0;JJ=1/2/C0*(u(1)-m0)^2;
for j=1:J
    JJ=JJ+1/2/sigma^2*(u(j+1)-alpha*sin(u(j)))^2;
    Phi=Phi+1/2/gamma^2*(y(j)-u(j+1))^2;
end
out=Phi+JJ;
```

5.3 Chapter 4 Programs

The programs `p8.m`–`p15.m`, used to generate the figures in Chapter 4, are presented in this section. Various filtering algorithms used to sample the posterior filtering distribution are given, involving both Gaussian approximation and particle approximation. Since these algorithms are run for very large times (large J), they will only be divided in two sections, `setup` in which the parameters are defined, and `solution` in which *both* the truth and observations are generated, *and* the online assimilation of the current observation into the filter solution is performed. The generation of truth can be separated into a `truth` section as in the previous sections, but two loops of length J would be required, and loops are inefficient in MATLAB , so the present format is preferred. The programs in this section are all very similar, and their output is also similar, giving rise to Figures 4.3-4.12. With the exception of `p8.m` and `p9.m`, the forward model is given by Example 2.8 (5.1), and the output is identical, given for `p10.m` through `p15.m` in Figures 4.5-4.7 and 4.8-4.10. Figures 4.11 and 4.12 compare the filters from the other Figures. `p8.m` features a two-dimensional linear forward model, and `p9.m` features the forward model from Example 2.9 (5.2). At the end of each program, the outputs are used to plot the mean and the covariance as well as the mean square error of the filter as functions of the iteration number j.

5.3.1. p8.m

The first filtering program is `p8.m` which contains an implementation of the Kalman Filter applied to Example 2.2:

$$v_{j+1} = Av_j + \xi_j, \quad \text{with} \quad A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

and observed data given by

$$y_{j+1} = Hv_{j+1} + \eta_{j+1}$$

with $H = (1, 0)$ and Gaussian noise. Thus only the first component of v_j is observed.

The parameters and initial condition are defined in the `setup` section, lines 3-19. The vectors `v`, `m` $\in \mathbb{R}^{N\times J}$, `y`$\in \mathbb{R}^J$, and `c` $\in \mathbb{R}^{N\times N\times J}$ are preallocated to hold the truth, mean, observations, and covariance over the `J` observation times defined in line 5. In particular, notice that the true initial condition is drawn from $N(m_0, C_0)$ in line 16, where $m_0 = 0$ and $C_0 = 1$ are defined in lines 10-11. The initial *estimate* of the distribution is defined in lines 17-18 as $N(m_0, C_0)$, where $m_0 \sim N(0, 100I)$ and $C_0 \leftarrow 100C_0$ so that the code may test the ability of the filter to lock onto the true distribution, asymptotically in j, given a poor initial estimate. That is to say, the values of (m_0, C_0) are *changed* such that the initial condition is *not* drawn from this distribution.

The main `solution` loop then follows in lines 21-34. The truth `v` and the data that are being assimilated `y` are sequentially generated within the loop, in lines 24-25. The filter prediction step, in lines 27-28, consists of computing the predictive mean and covariance $\widehat{m}_j$ and $\widehat{C}_j$ as defined in (4.4) and (4.5) respectively:

$$\widehat{m}_{j+1} = Am_j, \quad \widehat{C}_{j+1} = AC_jA^T + \Sigma.$$

Notice that indices are not used for the transient variables `mhat` and `chat` representing $\widehat{m}_j$ and $\widehat{C}_j$ because they will not be saved from one iteration to the next. In lines 30-33 we implement the analysis formulae for the Kalman filter from Corollary 4.2. In particular, the innovation between the observation of the predicted mean and the actual observation, as introduced in Corollary 4.2, is first computed in line 30

$$d_j = y_j - H\widehat{m}_j. \tag{5.12}$$

Again `d`, which represents d_j, does not have any index for the same reason as above. Next, the Kalman gain defined in Corollary 4.2 is computed in line 31

$$K_j = \widehat{C}_j H^T (H\widehat{C}_j H^T + \Gamma)^{-1}. \tag{5.13}$$

Once again index j is not used for the transient variable `K` representing K_j. Notice the "forward slash" `/` is used to compute `B/A=B A`$^{-1}$. This is an internal function of MATLAB which will analyze the matrices `B` and `A` to determine an "optimal" method for inversion, given their structure. The update given in Corollary 4.2 is completed in lines 30-32 with the equations

$$m_j = \widehat{m}_j + K_j d_j \quad \text{and} \quad C_j = (I - K_j H)\widehat{C}_j. \tag{5.14}$$

Finally, in lines 36-50 the outputs of the program are used to plot the mean and the covariance as well as the mean square error of the filter as functions of the iteration number j, as shown in Figure 4.3.

```
clear;set(0,’defaultaxesfontsize’,20);format long
%%% p8.m Kalman Filter, Ex. 1.2
%% setup

J=1e3;% number of steps
N=2;% dimension of state
I=eye(N);% identity operator
gamma=1;% observational noise variance is gamma^2*I
sigma=1;% dynamics noise variance is sigma^2*I
C0=eye(2);% prior initial condition variance
m0=[0;0];% prior initial condition mean
sd=10;rng(sd);% choose random number seed
A=[0 1;-1 0];% dynamics determined by A

m=zeros(N,J);v=m;y=zeros(J,1);c=zeros(N,N,J);% pre-allocate
v(:,1)=m0+sqrtm(C0)*randn(N,1);% initial truth
m(:,1)=10*randn(N,1);% initial mean/estimate
c(:,:,1)=100*C0;% initial covariance
H=[1,0];% observation operator

%% solution % assimilate!

for j=1:J
    v(:,j+1)=A*v(:,j) + sigma*randn(N,1);% truth
    y(j)=H*v(:,j+1)+gamma*randn;% observation

    mhat=A*m(:,j);% estimator predict
    chat=A*c(:,:,j)*A’+sigma^2*I;% covariance predict

    d=y(j)-H*mhat;% innovation
    K=(chat*H’)/(H*chat*H’+gamma^2);% Kalman gain
    m(:,j+1)=mhat+K*d;% estimator update
    c(:,:,j+1)=(I-K*H)*chat;% covariance update
end

figure;js=21;plot([0:js-1],v(2,1:js));hold;plot([0:js-1],m(2,1:js),’m’);
plot([0:js-1],m(2,1:js)+reshape(sqrt(c(2,2,1:js)),1,js),’r--’);
plot([0:js-1],m(2,1:js)-reshape(sqrt(c(2,2,1:js)),1,js),’r--’);
hold;grid;xlabel(’iteration, j’);
title(’Kalman Filter, Ex. 1.2’);

figure;plot([0:J],reshape(c(1,1,:)+c(2,2,:),J+1,1));hold
plot([0:J],cumsum(reshape(c(1,1,:)+c(2,2,:),J+1,1))./[1:J+1]’,’m’, ...
’Linewidth’,2); grid; hold;xlabel(’iteration, j’);axis([1 1000 0 50]);
title(’Kalman Filter Covariance, Ex. 1.2’);

figure;plot([0:J],sum((v-m).^2));hold;
plot([0:J],cumsum(sum((v-m).^2))./[1:J+1],’m’,’Linewidth’,2);grid
hold;xlabel(’iteration, j’);axis([1 1000 0 50]);
title(’Kalman Filter Error, Ex. 1.2’)
```

5.3.2. p9.m

The program `p9.m` contains an implementation of the 3DVAR method applied to the chaotic logistic map of Example 2.4 (5.2) for $r = 4$. As in the previous section, the parameters and initial condition are defined in the `setup` section, lines 3-16. In particular, notice that the truth initial condition `v(1)` and initial mean `m(1)`, are now initialized in lines 12-13 with a *uniform* random number using the command `rand`, so that they are in the interval $[0, 1]$ where the model is well-defined. Indeed the solution will eventually become unbounded if initial conditions are chosen outside this interval. With this in mind, we set the dynamics noise `sigma = 0` in line 8, i.e. deterministic dynamics, so that the true dynamics themselves do not go unbounded.

The analysis step of 3DVAR consists of minimizing

$$\mathsf{I}_{\text{filter}}(v) = \frac{1}{2}|\Gamma^{-\frac{1}{2}}(y_{j+1} - Hv)|^2 + \frac{1}{2}|\widehat{C}^{-\frac{1}{2}}(v - \Psi(m_j))|^2.$$

In this one-dimensional case we set $\Gamma = \gamma^2$, $\widehat{C} = \sigma^2$ and define $\eta^2 = \gamma^2/\eta^2$. The stabilization parameter η (`eta`) from Example 4.12 is set in line 14, representing the ratio in uncertainty in the data to that of the model; equivalently it measures trust in the model over the observations. The choice $\eta = 0$ means the model is irrelevant in the minimization step (4.12) of 3DVAR , in the observed space –the synchronization filter. Since, in the example, the signal space and observation space both have dimension equal to one, the choice $\eta = 0$ simply corresponds to using only the data. In contrast the choice $\eta = \infty$ ignores the observations and uses only the model.

The 3DVAR set-up gives rise to the constant scalar covariance `C` and resultant constant scalar gain `K`; this should not be confused with the changing K_j in (5.13), temporarily defined by `K` in line 31 of `p8.m`. The main `solution` loop follows in lines 20-33. Up to the different forward model, lines 21-22, 24, 26, and 27 of this program are identical to lines 24-25, 27, 30, and 32 of `p8.m` described in section 5.3.1. The only other difference is that the covariance updates are not here because of the constant covariance assumption underlying the 3DVAR algorithm.

The 3DVAR filter may in principle generate estimated mean `mhat` outside $[0, 1]$, because of the noise in the data. In order to flag potential unbounded trajectories of the filter, which in principle could arise because of this, an extra stopping criteria is included in lines 29-32. To illustrate this try setting `sigma`$\neq 0$ in line 8. Then the signal will eventually become unbounded, regardless of how small the noise variance is chosen. In this case the estimate will surely blowup while tracking the unbounded signal. Otherwise, if η is chosen appropriately so as to stabilize the filter it is extremely unlikely that the estimate will ever blowup. Finally, similarly to `p8.m`, in the last lines of the program we use the outputs of the program in order to produce Figure 4.4, namely plotting the mean and the covariance as well as the mean square error of the filter as functions of the iteration number j.

```
clear;set(0,'defaultaxesfontsize',20);format long
%%% p9.m 3DVAR Filter, deterministic logistic map (Ex. 1.4)
%% setup

J=1e3;% number of steps
r=4;% dynamics determined by r
gamma=1e-1;% observational noise variance is gamma^2
sigma=0;% dynamics noise variance is sigma^2
sd=10;rng(sd);% choose random number seed

m=zeros(J,1);v=m;y=m;% pre-allocate
v(1)=rand;% initial truth, in [0,1]
m(1)=rand;% initial mean/estimate, in [0,1]
eta=2e-1;% stabilization coefficient 0 < eta << 1
C=gamma^2/eta;H=1;% covariance and observation operator
K=(C*H')/(H*C*H'+gamma^2);% Kalman gain

%% solution % assimilate!

for j=1:J
    v(j+1)=r*v(j)*(1-v(j)) + sigma*randn;% truth
    y(j)=H*v(j+1)+gamma*randn;% observation

    mhat=r*m(j)*(1-m(j));% estimator predict

    d=y(j)-H*mhat;% innovation
    m(j+1)=mhat+K*d;% estimator update

    if norm(mhat)>1e5
        disp('blowup!')
        break
    end
end
js=21;% plot truth, mean, standard deviation, observations
figure;plot([0:js-1],v(1:js));hold;plot([0:js-1],m(1:js),'m');
plot([0:js-1],m(1:js)+sqrt(C),'r--');plot([1:js-1],y(1:js-1),'kx');
plot([0:js-1],m(1:js)-sqrt(C),'r--');hold;grid;xlabel('iteration, j');
title('3DVAR Filter, Ex. 1.4')

figure;plot([0:J],C*[0:J].^0);hold
plot([0:J],C*[0:J].^0,'m','Linewidth',2);grid
hold;xlabel('iteration, j');title('3DVAR Filter Covariance, Ex. 1.4');

figure;plot([0:J],(v-m).^2);hold;
plot([0:J],cumsum((v-m).^2)./[1:J+1]','m','Linewidth',2);grid
hold;xlabel('iteration, j');
title('3DVAR Filter Error, Ex. 1.4')
```

5.3.3. p10.m

A variation of program `p9.m` is given by `p10.m`, where the 3DVAR filter is implemented for Example 2.3 given by (5.1). Indeed the remaining programs of this section will all be for the same Example 2.3 so this will not be mentioned again. In this case, the initial condition is again taken as a draw from the prior $N(m_0, C_0)$ as in `p7.m`, and the initial mean estimate is again *changed* to $m_0 \sim N(0, 100I)$ so that the code may test the ability of the filter to lock onto the signal given a poor initial estimate. Furthermore, for this problem there is no need to introduce the stopping criteria present in the case of `p9.m` since the underlying deterministic dynamics are dissipative. The output of this program is shown in Figure 4.5.

```
clear;set(0,’defaultaxesfontsize’,20);format long
%%% p10.m 3DVAR Filter, sin map (Ex. 1.3)
%% setup

J=1e3;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=3e-1;% dynamics noise variance is sigma^2
C0=9e-2;% prior initial condition variance
m0=0;% prior initial condition mean
sd=1;rng(sd);% choose random number seed

m=zeros(J,1);v=m;y=m;% pre-allocate
v(1)=m0+sqrt(C0)*randn;% initial truth
m(1)=10*randn;% initial mean/estimate
eta=2e-1;% stabilization coefficient 0 < eta << 1
c=gamma^2/eta;H=1;% covariance and observation operator
K=(c*H’)/(H*c*H’+gamma^2);% Kalman gain

%% solution % assimilate!

for j=1:J
    v(j+1)=alpha*sin(v(j)) + sigma*randn;% truth
    y(j)=H*v(j+1)+gamma*randn;% observation

    mhat=alpha*sin(m(j));% estimator predict

    d=y(j)-H*mhat;% innovation
    m(j+1)=mhat+K*d;% estimator update

end

js=21;% plot truth, mean, standard deviation, observations
figure;plot([0:js-1],v(1:js));hold;plot([0:js-1],m(1:js),’m’);
plot([0:js-1],m(1:js)+sqrt(c),’r--’);plot([1:js-1],y(1:js-1),’kx’);
plot([0:js-1],m(1:js)-sqrt(c),’r--’);hold;grid;xlabel(’iteration, j’);
title(’3DVAR Filter, Ex. 1.3’)

figure;plot([0:J],c*[0:J].^0);hold
plot([0:J],c*[0:J].^0,’m’,’Linewidth’,2);grid
hold;xlabel(’iteration, j’);
title(’3DVAR Filter Covariance, Ex. 1.3’);

figure;plot([0:J],(v-m).^2);hold;
plot([0:J],cumsum((v-m).^2)./[1:J+1]’,’m’,’Linewidth’,2);grid
hold;xlabel(’iteration, j’);
title(’3DVAR Filter Error, Ex. 1.3’)
```

5.3.4. p11.m

The next program is `p11.m`. This program comprises an implementation of the extended Kalman Filter.It is very similar in structure to `p8.m`, except with a different forward model. Since the dynamics are scalar, the observation operator is defined by setting `H` to take value 1 in line 16. The predicting covariance $\widehat{C}_j$ is not independent of the mean as it is for the linear problem `p8.m`. Instead, as described in section 4.2.2, it is determined via the *linearization* of the forward map around m_j, in line 26:

$$\widehat{C}_{j+1} = (\alpha \cos(m_j))\, C_j\, (\alpha \cos(m_j))\,.$$

As in `p8.m` we *change* the prior to a poor initial estimate of the distribution to study if, and how, the filter locks onto a neighbourhood of the true signal, despite poor initialization, for large j. This initialization is in lines 15-16, where $m_0 \sim N(0, 100I)$ and $C_0 \leftarrow 10C_0$. Subsequent filtering programs use an identical initialization, with the same rationale as in this case. We will not state this again. The output of this program is shown in Figure 4.6.

```
clear;set(0,’defaultaxesfontsize’,20);format long
%%% p11.m  Extended Kalman Filter, sin map (Ex. 1.3)
%% setup

J=1e3;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=3e-1;% dynamics noise variance is sigma^2
C0=9e-2;% prior initial condition variance
m0=0;% prior initial condition mean
sd=1;rng(sd);% choose random number seed

m=zeros(J,1);v=m;y=m;c=m;% pre-allocate
v(1)=m0+sqrt(C0)*randn;% initial truth
m(1)=10*randn;% initial mean/estimate
c(1)=10*C0;H=1;% initial covariance and observation operator

%% solution % assimilate!

for j=1:J

    v(j+1)=alpha*sin(v(j)) + sigma*randn;% truth
    y(j)=H*v(j+1)+gamma*randn;% observation

    mhat=alpha*sin(m(j));% estimator predict
    chat=alpha*cos(m(j))*c(j)*alpha*cos(m(j))+sigma^2;% covariance predict

    d=y(j)-H*mhat;% innovation
    K=(chat*H’)/(H*chat*H’+gamma^2);% Kalman gain
    m(j+1)=mhat+K*d;% estimator update
    c(j+1)=(1-K*H)*chat;% covariance update

end

js=21;% plot truth, mean, standard deviation, observations
figure;plot([0:js-1],v(1:js));hold;plot([0:js-1],m(1:js),’m’);
plot([0:js-1],m(1:js)+sqrt(c(1:js)),’r--’);plot([1:js-1],y(1:js-1),’kx’);
plot([0:js-1],m(1:js)-sqrt(c(1:js)),’r--’);hold;grid;
xlabel(’iteration, j’);title(’ExKF, Ex. 1.3’)

figure;plot([0:J],c);hold
plot([0:J],cumsum(c)./[1:J+1]’,’m’,’Linewidth’,2);grid
hold;xlabel(’iteration, j’);
title(’ExKF Covariance, Ex. 1.3’);

figure;plot([0:J],(v-m).^2);hold;
plot([0:J],cumsum((v-m).^2)./[1:J+1]’,’m’,’Linewidth’,2);grid
hold;xlabel(’iteration, j’);
title(’ExKF Error, Ex. 1.3’)
```

5.3.5. p12.m

The program `p12.m` contains an implementation of the ensemble Kalman Filter, with perturbed observations, as described in section 4.2.3.The structure of this program is again very similar to `p8.m` and `p11.m`, except now an ensemble of particles, of size `N` defined in line 12, is retained as an approximation of the filtering distribution. The ensemble $\{v^{(n)}\}_{n=1}^N$ represented by the matrix `U` is then constructed out of draws from this Gaussian in line 18, and the mean m_0' is reset to the ensemble sample mean.

In line 27 the predicting ensemble $\{\widehat{v}_j^{(n)}\}_{n=1}^N$ represented by the matrix `Uhat` is computed from a realization of the forward map applied to each ensemble member. This is then used to compute the ensemble sample mean $\widehat{m}_j$ (`mhat`) and covariance $\widehat{C}_j$ (`chat`). There is now an ensemble of "innovations" with a new i.i.d. realization $y_j^{(n)} \sim N(y_j, \Gamma)$ for each ensemble member, computed in line 31(not to be confused with the actual innovation as defined in Equation (5.12))

$$d_j^{(n)} = y_j^{(n)} - H\widehat{v}_j^{(n)}.$$

The Kalman gain K_j (`K`) is computed using (5.13), very similarly to `p8.m` and `p11.m`, and the ensemble of updates are computed in line 33:

$$v_j^{(n)} = \widehat{v}_j^{(n)} + K_j d_j^{(n)}.$$

The output of this program is shown in Figure 4.7. Furthermore, long simulations of length $J = 10^5$ were performed for this and the previous two programs `p10.m` and `p11.m` and their errors are compared in Figure 4.11.

```
clear;set(0,'defaultaxesfontsize',20);format long
%%% p12.m Ensemble Kalman Filter (PO), sin map (Ex. 1.3)
%% setup

J=1e5;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=3e-1;% dynamics noise variance is sigma^2
C0=9e-2;% prior initial condition variance
m0=0;% prior initial condition mean
sd=1;rng(sd);% choose random number seed
N=10;% number of ensemble members

m=zeros(J,1);v=m;y=m;c=m;U=zeros(J,N);% pre-allocate
v(1)=m0+sqrt(C0)*randn;% initial truth
m(1)=10*randn;% initial mean/estimate
c(1)=10*C0;H=1;% initial covariance and observation operator
U(1,:)=m(1)+sqrt(c(1))*randn(1,N);m(1)=sum(U(1,:))/N;% initial ensemble

%% solution % assimilate!

for j=1:J

    v(j+1)=alpha*sin(v(j)) + sigma*randn;% truth
    y(j)=H*v(j+1)+gamma*randn;% observation

    Uhat=alpha*sin(U(j,:))+sigma*randn(1,N);% ensemble predict
    mhat=sum(Uhat)/N;% estimator predict
    chat=(Uhat-mhat)*(Uhat-mhat)'/(N-1);% covariance predict

    d=y(j)+gamma*randn(1,N)-H*Uhat;% innovation
    K=(chat*H')/(H*chat*H'+gamma^2);% Kalman gain
    U(j+1,:)=Uhat+K*d;% ensemble update
    m(j+1)=sum(U(j+1,:))/N;% estimator update
    c(j+1)=(U(j+1,:)-m(j+1))*(U(j+1,:)-m(j+1))'/(N-1);% covariance update

end

js=21;% plot truth, mean, standard deviation, observations
figure;plot([0:js-1],v(1:js));hold;plot([0:js-1],m(1:js),'m');
plot([0:js-1],m(1:js)+sqrt(c(1:js)),'r--');plot([1:js-1],y(1:js-1),'kx');
plot([0:js-1],m(1:js)-sqrt(c(1:js)),'r--');hold;grid;
xlabel('iteration, j');title('EnKF, Ex. 1.3')

figure;plot([0:J],c);hold
plot([0:J],cumsum(c)./[1:J+1]','m','Linewidth',2);grid
hold;xlabel('iteration, j');
title('EnKF Covariance, Ex. 1.3');

figure;plot([0:J],(v-m).^2);hold;
plot([0:J],cumsum((v-m).^2)./[1:J+1]','m','Linewidth',2);grid
hold;xlabel('iteration, j');
title('EnKF Error, Ex. 1.3')
```

5.3.6. p13.m

The program `p13.m` contains a particular square-root filter implementation of the ensemble Kalman filter , namely the ETKF filter described in detail in section 4.2.4. The program thus is very similar to `p12.m` for the EnKF with perturbed observations. In particular, the filtering distribution of the state is again approximated by an ensemble of particles. The predicting ensemble $\{\widehat{v}_j^{(n)}\}_{n=1}^N$ (`Uhat`), mean $\widehat{m}_j$(`mhat`), and covariance $\widehat{C}_j$ (`chat`) are computed exactly as in `p12.m`. However, this time the covariance is kept in factorized form $\widehat{X}_j\widehat{X}_j^\top = \widehat{C}_j$ in lines 29-30, with factors denoted `Xhat`. The transformation matrix is computed in line 31

$$T_j = \left(I_N + \widehat{X}_j^\top H^\top \Gamma^{-1} H \widehat{X}_j\right)^{-\frac{1}{2}},$$

and $X_j = \widehat{X}_j T_j$ (`X`) is computed in line 32, from which the covariance $C_j = X_j X_j^\top$ is reconstructed in line 38. A single innovation d_j is computed in line 34 and a single updated mean m_j is then computed in line 36 using the Kalman gain K_j (5.13) computed in line 35. This is the same as in the Kalman Filter and extended Kalman filter (ExKF) of `p8.m` and `p11.m`, in contrast to the EnKF with perturbed observations appearing in `p12.m`. The ensemble is then updated to `U` in line 37 using the formula

$$v_j^{(n)} = m_j + X_j^{(n)}\sqrt{N-1},$$

where $X_j^{(n)}$ is the n^{th} column of X_j.

Notice that the operator which is factorized and inverted has dimension N, which in this case is large in comparison to the state and observation dimensions. This is of course natural for computing sample statistics but in the context of the one dimensional examples considered here makes `p13.m` run far more slowly than `p12.m`. However in many applications the signal state-space dimension is the largest, with the observation dimension coming next, and the ensemble size being far smaller than either of these. In this context the ETKF has become a very popular method. So its relative inefficiency, compared for example with the perturbed observations Kalman filter, should not be given too much weight in the overall evaluation of the method. Results illustrating the algorithm are shown in Figure 4.8.

```
clear;set(0,’defaultaxesfontsize’,20);format long
%%% p13.m Ensemble Kalman Filter (ETKF), sin map (Ex. 1.3)
%% setup

J=1e3;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=3e-1;% dynamics noise variance is sigma^2
C0=9e-2;% prior initial condition variance
m0=0;% prior initial condition mean
sd=1;rng(sd);% choose random number seed
N=10;% number of ensemble members

m=zeros(J,1);v=m;y=m;c=m;U=zeros(J,N);% pre-allocate
v(1)=m0+sqrt(C0)*randn;% initial truth
m(1)=10*randn;% initial mean/estimate
c(1)=10*C0;H=1;% initial covariance and observation operator
U(1,:)=m(1)+sqrt(c(1))*randn(1,N);m(1)=sum(U(1,:))/N;% initial ensemble

%% solution % assimilate!

for j=1:J

    v(j+1)=alpha*sin(v(j)) + sigma*randn;% truth
    y(j)=H*v(j+1)+gamma*randn;% observation

    Uhat=alpha*sin(U(j,:))+sigma*randn(1,N);% ensemble predict
    mhat=sum(Uhat)/N;% estimator predict
    Xhat=(Uhat-mhat)/sqrt(N-1);% centered ensemble
    chat=Xhat*Xhat’;% covariance predict
    T=sqrtm(inv(eye(N)+Xhat’*H’*H*Xhat/gamma^2));% sqrt transform
    X=Xhat*T;% transformed centered ensemble

    d=y(j)-H*mhat;randn(1,N);% innovation
    K=(chat*H’)/(H*chat*H’+gamma^2);% Kalman gain
    m(j+1)=mhat+K*d;% estimator update
    U(j+1,:)=m(j+1)+X*sqrt(N-1);% ensemble update
    c(j+1)=X*X’;% covariance update

end

js=21;% plot truth, mean, standard deviation, observations
figure;plot([0:js-1],v(1:js));hold;plot([0:js-1],m(1:js),’m’);
plot([0:js-1],m(1:js)+sqrt(c(1:js)),’r--’);plot([1:js-1],y(1:js-1),’kx’);
plot([0:js-1],m(1:js)-sqrt(c(1:js)),’r--’);hold;grid;
xlabel(’iteration, j’);title(’EnKF(ETKF), Ex. 1.3’);

figure;plot([0:J],(v-m).^2);hold;
plot([0:J],cumsum((v-m).^2)./[1:J+1]’,’m’,’Linewidth’,2);grid
plot([0:J],cumsum(c)./[1:J+1]’,’r--’,’Linewidth’,2);
hold;xlabel(’iteration, j’);
title(’EnKF(ETKF) Error, Ex. 1.3’)
```

5.3.7. p14.m

The program `p14.m` is an implementation of the standard SIRS filter from subsection 4.3.2. The `setup` section is almost identical to the EnKF methods, because those methods also rely on particle approximations of the filtering distribution . However, the particle filters consistently estimate quite general distributions, whilst the EnKF is only provably accurate for Gaussian distributions. The truth and data generation and ensemble prediction in lines 24-27 are the same as in `p12.m` and `p13.m`. The way this prediction in line 27 is phrased in section 4.3.2 is $\widehat{v}_{j+1}^{(n)} \sim \mathbb{P}(\cdot|v_j^{(n)})$. An ensemble of "innovation" terms $\{d_j^{(n)}\}_{n=1}^N$ are again required again, but all using the *same* observation, as computed in line 28. Assuming $w_j^{(n)} = 1/N$, then

$$\widehat{w}_j^{(n)} \propto \mathbb{P}(y_j|v_j^{(n)}) \propto \exp\left\{-\frac{1}{2}\left|d_j^{(n)}\right|_\Gamma^2\right\},$$

where $d_j^{(n)}$ is the innovation of the n^{th} particle, as given in (4.27). The vector of un-normalized weights $\{\widehat{w}_j^{(n)}\}_{n=1}^N$ (`what`) are computed in line 29 and normalized to $\{w_j^{(n)}\}_{n=1}^N$ (`w`) in line 30. Lines 32-39 implement the resampling step. First, the cumulative distribution function of the weights $W \in [0,1]^N$ (`ws`) is computed in line 32. Notice W has the properties that $W_1 = w_j^{(1)}$, $W_n \leq W_{n+1}$, and $W_N = 1$. Then N uniform random numbers $\{u^{(n)}\}_{n=1}^N$ are drawn. For each $u^{(n)}$, let n^* be such that $W_{n^*-1} \leq u^{(n)} < W_{n^*}$. This n^* (`ix`) is found in line 34 using the `find` function, which can identify the first or last element in an array to exceed zero (see `help` file): `ix = find ( ws > rand, 1, 'first' )`. This corresponds to drawing the $(n^*)^{th}$ element from the discrete measure defined by $\{w_j^{(n)}\}_{n=1}^N$. The n^{th} particle $v_j^{(n)}$ (`U(j+1,n)`) is set to be equal to $\widehat{v}_j^{(n^*)}$ (`Uhat(ix)`) in line 37. The sample mean and covariance are then computed in lines 41-42. The rest of the program follows the others, generating the output displayed in Figure 4.9.

```
clear;set(0,’defaultaxesfontsize’,20);format long
%%% p14.m Particle Filter (SIRS), sin map (Ex. 1.3)
%% setup

J=1e3;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=3e-1;% dynamics noise variance is sigma^2
C0=9e-2;% prior initial condition variance
m0=0;% prior initial condition mean
sd=1;rng(sd);% choose random number seed
N=100;% number of ensemble members

m=zeros(J,1);v=m;y=m;c=m;U=zeros(J,N);% pre-allocate
v(1)=m0+sqrt(C0)*randn;% initial truth
m(1)=10*randn;% initial mean/estimate
c(1)=10*C0;H=1;% initial covariance and observation operator
U(1,:)=m(1)+sqrt(c(1))*randn(1,N);m(1)=sum(U(1,:))/N;% initial ensemble

%% solution % Assimilate!
for j=1:J
    v(j+1)=alpha*sin(v(j)) + sigma*randn;% truth
    y(j)=H*v(j+1)+gamma*randn;% observation

    Uhat=alpha*sin(U(j,:))+sigma*randn(1,N);% ensemble predict
    d=y(j)-H*Uhat;% ensemble innovation
    what=exp(-1/2*(1/gamma^2*d.^2));% weight update
    w=what/sum(what);% normalize predict weights

    ws=cumsum(w);% resample: compute cdf of weights
    for n=1:N
        ix=find(ws>rand,1,’first’);% resample: draw rand \sim U[0,1] and
        % find the index of the particle corresponding to the first time
        % the cdf of the weights exceeds rand.
        U(j+1,n)=Uhat(ix);% resample: reset the nth particle to the one
        % with the given index above
    end

    m(j+1)=sum(U(j+1,:))/N;% estimator update
    c(j+1)=(U(j+1,:)-m(j+1))*(U(j+1,:)-m(j+1))’/N;% covariance update
end

js=21;% plot truth, mean, standard deviation, observations
figure;plot([0:js-1],v(1:js));hold;plot([0:js-1],m(1:js),’m’);
plot([0:js-1],m(1:js)+sqrt(c(1:js)),’r--’);plot([1:js-1],y(1:js-1),’kx’);
plot([0:js-1],m(1:js)-sqrt(c(1:js)),’r--’);hold;grid;
xlabel(’iteration, j’);title(’Particle Filter (Standard), Ex. 1.3’);

figure;plot([0:J],(v-m).^2);hold;
plot([0:J],cumsum((v-m).^2)./[1:J+1]’,’m’,’Linewidth’,2);grid
hold;xlabel(’iteration, j’);
title(’Particle Filter (Standard) Error, Ex. 1.3’)
```

5.3.8. p15.m

The program `p15.m` is an implementation of the SIRS(OP) algorithm from subsection 4.3.3. The `setup` section and truth and observation generation are again the same as in the previous programs. The difference between this program and `p14.m` arises arises because the importance sampling proposal kernel Q_j with density $\mathbb{P}(v_{j+1}|v_j, y_{j+1})$ used to propose each $\widehat{v}_{j+1}^{(n)}$ given each particular $v_j^{(n)}$; in particular Q_j depends on the next data point whereas the kernel P used in `p14.m` has density $\mathbb{P}(v_{j+1}|v_j)$ which is independent of y_{j+1}.

Observe that if $v_j^{(n)}$ and y_{j+1} are both fixed, then $\mathbb{P}\left(v_{j+1}|v_j^{(n)}, y_{j+1}\right)$ is the density of the Gaussian with mean $m'^{(v)}$ and covariance Σ' given by

$$m'^{(n)} = \Sigma'\left(\Sigma^{-1}\Psi\left(v_j^{(n)}\right) + H^\top\Gamma^{-1}y_{j+1}\right), \quad (\Sigma')^{-1} = \Sigma^{-1} + H^\top\Gamma^{-1}H.$$

Therefore, Σ' (`Sig`) and the ensemble of means $\left\{m'^{(n)}\right\}_{n=1}^N$ (vector `em`) are computed in lines 27 and 28 and used to sample $\widehat{v}_{j+1}^{(n)} \sim N(m'^{(n)}, \Sigma')$ in line 29 for all of $\left\{\widehat{v}_{j+1}^{(n)}\right\}_{n=1}^N$ (`Uhat`).

Now the weights are therefore updated by (4.34) rather than (4.27), i.e. assuming $w_j^{(n)} = 1/N$, then

$$\widehat{w}_{j+1}^{(n)} \propto \mathbb{P}\left(y_{j+1}|v_j^{(n)}\right) \propto \exp\left\{-\frac{1}{2}\left|y_{j+1} - \Psi\left(v_j^{(n)}\right)\right|_{\Gamma+\Sigma}^2\right\}.$$

This is computed in lines 31-32, using another auxiliary "innovation" vector `d` in line 31. Lines 35-45 are again identical to lines 32-42 of program `p14.m`, performing the resampling step and computing sample mean and covariance.

The output of this program was used to produce Figure 4.10 similar to the other filtering algorithms . Furthermore, long simulations of length $J = 10^5$ were performed for this and the previous three programs `p12.m`, `p13.m` and `p14.m` and their errors are compared in Figure 4.12, similarly to Figure 4.11 comparing the basic filters `p10.m`, `p11.m`, and `p12.m`.

```
clear;set(0,’defaultaxesfontsize’,20);format long
%%% p15.m Particle Filter (SIRS, OP), sin map (Ex. 1.3)
%% setup

J=1e3;% number of steps
alpha=2.5;% dynamics determined by alpha
gamma=1;% observational noise variance is gamma^2
sigma=3e-1;% dynamics noise variance is sigma^2
C0=9e-2;% prior initial condition variance
m0=0;% prior initial condition mean
sd=1;rng(sd);% choose random number seed
N=100;% number of ensemble members

m=zeros(J,1);v=m;y=m;c=m;U=zeros(J,N);% pre-allocate
v(1)=m0+sqrt(C0)*randn;% initial truth
m(1)=10*randn;% initial mean/estimate
c(1)=10*C0;H=1;% initial covariance and observation operator
U(1,:)=m(1)+sqrt(c(1))*randn(1,N);m(1)=sum(U(1,:))/N;% initial ensemble

%% solution % Assimilate!
for j=1:J
    v(j+1)=alpha*sin(v(j)) + sigma*randn;% truth
    y(j)=H*v(j+1)+gamma*randn;% observation

    Sig=inv(inv(sigma^2)+H’*inv(gamma^2)*H);% optimal proposal covariance
    em=Sig*(inv(sigma^2)*alpha*sin(U(j,:))+H’*inv(gamma^2)*y(j));% mean
    Uhat=em+sqrt(Sig)*randn(1,N);% ensemble optimally importance sampled

    d=y(j)-H*alpha*sin(U(j,:));% ensemble innovation
    what=exp(-1/2/(sigma^2+gamma^2)*d.^2);% weight update
    w=what/sum(what);% normalize predict weights

    ws=cumsum(w);% resample: compute cdf of weights
    for n=1:N
        ix=find(ws>rand,1,’first’);% resample: draw rand \sim U[0,1] and
        % find the index of the particle corresponding to the first time
        % the cdf of the weights exceeds rand.
        U(j+1,n)=Uhat(ix);% resample: reset the nth particle to the one
        % with the given index above
    end

    m(j+1)=sum(U(j+1,:))/N;% estimator update
    c(j+1)=(U(j+1,:)-m(j+1))*(U(j+1,:)-m(j+1))’/N;% covariance update
end

js=21;%plot truth, mean, standard deviation, observations
figure;plot([0:js-1],v(1:js));hold;plot([0:js-1],m(1:js),’m’);
plot([0:js-1],m(1:js)+sqrt(c(1:js)),’r--’);plot([1:js-1],y(1:js-1),’kx’);
plot([0:js-1],m(1:js)-sqrt(c(1:js)),’r--’);hold;grid;
xlabel(’iteration, j’);title(’Particle Filter (Optimal), Ex. 1.3’);
```

5.4 ODE Programs

The programs `p16.m` and `p17.m` are used to simulate and plot the Lorenz '63 and '96 models from Examples 2.6 and 2.7, respectively. These programs are both MATLAB functions, similar to the program `p7.m` presented in Section 5.2.5. The reason for using functions and not scripts is that the black box MATLAB built-in function `ode45` can be used for the time integration (see `help` page for details regarding this function). Therefore, each has an *auxiliary* function defining the right-hand side of the given ODE, which is passed via *function handle* to `ode45`.

5.4.1. p16.m

The first of the ODE programs, `p16.m`, integrates the Lorenz '63 model 2.6. The setup section of the program, on lines 4-11, defines the parameters of the model and the initial conditions. In particular, a random Gaussian initial condition is chosen in line 9, and a small perturbation to its first (x) component is introduced in line 10. The trajectories are computed on lines 13-14 using the built-in function `ode45`. Notice that the auxiliary function `lorenz63`, defined on line 29, takes as arguments (t, y), prescribed through the definition of the function handle `@(t,y)`, while (α, b, r) are given as fixed parameters `(a,b,r)`, defining the particular instance of the function. The argument t is intended for defining *non-autonomous* ODE, and is spurious here as it is an *autonomous* ODE and therefore t does not appear on the right-hand side. It is nonetheless included for completeness, and causes no harm. The Euclidean norm of the error is computed in line 16, and the results are plotted similarly to previous programs in lines 18-25. This program is used to plot Figs. 2.6 and 2.7.

5.4.2. p17.m

The second of the ODE programs, `p17.m`, integrates the `J=40` dimensional Lorenz '96 model 2.7. This program is almost identical to the previous one, where a small perturbation of the random Gaussian initial condition defined on line 9 is introduced on lines 10-11. The major difference is the function passed to `ode45` on lines 14-15, which now defines the right-hand side of the Lorenz '96 model given by sub-function `lorenz96` on line 30. Again the system is autonomous, and the spurious t variable is included for completeness. A few of the 40 degrees of freedom are plotted along with the error in lines 19-27. This program is used to plot Figs. 2.8 and 2.9

```
function this=p16
clear;set(0,’defaultaxesfontsize’,20);format long
%%% p16.m Lorenz ’63 (Ex. 2.6)
%% setup

a=10;b=8/3;r=28;% define parameters
sd=1;rng(sd);% choose random number seed

initial=randn(3,1);% choose initial condition
initial1=initial + [0.0001;0;0];% choose perturbed initial condition

%% calculate the trajectories with blackbox
[t1,y]=ode45(@(t,y) lorenz63(t,y,a,b,r), [0 100], initial);
[t,y1]=ode45(@(t,y) lorenz63(t,y,a,b,r), t1, initial1);

error=sqrt(sum((y-y1).^2,2));% calculate error

%% plot results

figure(1), semilogy(t,error,’k’)
axis([0 100 10^-6 10^2])
set(gca,’YTick’,[10^-6 10^-4 10^-2 10^0 10^2])

figure(2), plot(t,y(:,1),’k’)
axis([0 100 -20 20])


%% auxiliary dynamics function definition
function rhs=lorenz63(t,y,a,b,r)

rhs(1,1)=a*(y(2)-y(1));
rhs(2,1)=-a*y(1)-y(2)-y(1)*y(3);
rhs(3,1)=y(1)*y(2)-b*y(3)-b*(r+a);
```

```
function this=p17
clear;set(0,'defaultaxesfontsize',20);format long
%%% p17.m Lorenz '96 (Ex. 2.7)
%% setup

J=40;F=8;% define parameters
sd=1;rng(sd);% choose random number seed

initial=randn(J,1);% choose initial condition
initial1=initial;
initial1(1)=initial(1)+0.0001;% choose perturbed initial condition

%% calculate the trajectories with blackbox
[t1,y]=ode45(@(t,y) lorenz96(t,y,F), [0 100], initial);
[t,y1]=ode45(@(t,y) lorenz96(t,y,F), t1, initial1);

error=sqrt(sum((y-y1).^2,2));% calculate error

%% plot results

figure(1), plot(t,y(:,1),'k')
figure(2), plot(y(:,1),y(:,J),'k')
figure(3), plot(y(:,1),y(:,J-1),'k')

figure(4), semilogy(t,error,'k')
axis([0 100 10^-6 10^2])
set(gca,'YTick',[10^-6 10^-4 10^-2 10^0 10^2])

%% auxiliary dynamics function definition
function rhs=lorenz96(t,y,F)

rhs=[y(end);y(1:end-1)].*([y(2:end);y(1)] - ...
    [y(end-1:end);y(1:end-2)]) - y + F*y.^0;
```

Chapter 6

Continuous Time: Formulation

6.1 Set-Up

6.1.1. Derivation of the Continuous Time Model

6.1.2. Stochastic Integration

6.2 Guiding Examples

6.3 Smoothing Problem

6.3.1. Probabilistic Formulation of Data Assimilation

6.3.2. Girsanov Formula

6.3.3. Conditional Probabilities

6.3.4. Stochastic Dynamics

6.3.5. Deterministic Dynamics

6.4 Filtering Problem

6.5 Illustrations

6.6 Bibliographic Notes

6.7 Exercises

Chapter 7

Continuous Time: Smoothing Algorithms

7.1 Linear Gaussian Problems: The Kalman-Bucy Smoother

7.2 Markov Chain-Monte Carlo Methods

7.2.1. Deterministic Dynamics

7.2.2. Stochastic Dynamics

7.3 Variational Methods

7.4 Illustrations

7.5 Bibliographic Notes

7.6 Exercises

Chapter 8

Continuous Time: Filtering Algorithms

8.1 Linear Gaussian Problems: The Kalman-Bucy Filter

8.2 Approximate Gaussian Filters

8.2.1. 3DVAR

8.2.2. Extended Kalman Filter

8.2.3. Ensemble Kalman Filter

8.3 The Particle Filter

8.4 Large-Time Behaviour of Filters

8.4.1. The Kalman-Bucy Filter in One Dimension

8.4.2. The 3DVAR Filter

8.5 Illustrations

8.6 Bibliographic Notes

8.7 Exercises

Chapter 9

Continuous Time: MATLAB Programs

9.1 Chapter 6 Programs

9.1.1. p1c.m

9.1.2. p2c.m

9.1.3. p3c.m

9.2 Chapter 7 Programs

9.2.1. p4c.m

9.2.2. p5c.m

9.2.3. p6c.m

9.2.4. p7c.m

9.3 Chapter 8 Programs

9.3.1. p8c.m

9.3.2. p10c.m

9.3.3. p11c.m

9.3.4. p12c.m

9.3.5. p13c.m

9.3.6. p14c.m

9.3.7. p15c.m

9.3.8. p16c.m

9.3.9. p17c.m

Bibliography

[1] H. Abarbanel. *Predicting the Future: Completing Models of Observed Complex Systems.* Springer, 2013.

[2] J. Anderson. A method for producing and evaluating probabilistic forecasts from ensemble model integrations. *Journal of Climate*, 9(7):1518–1530, 1996.

[3] J. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Mon. Wea. Rev.*, 129(12):2884–2903, 2001.

[4] J. Anderson and S. Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Mon. Wea. Rev.*, 127(12):2741–2758, 1999.

[5] A. Andrews. A square root formulation of the Kalman covariance equations. *AIAA Journal*, 6(6):1165–1166, 1968.

[6] A. Apte, C. K. R. T. Jones, A. M. Stuart, and J. Voss. Data assimilation: mathematical and statistical perspectives. *Int. J. Num. Meth. Fluids*, 56:1033–1046, 2008.

[7] L. Arnold. *Random Dynamical Systems.* Springer Monographs in Mathematics. Springer-Verlag, Berlin, 1998.

[8] V. Baladi. *Positive Transfer Operators and Decay of Correlations.* World Scientific, 2000.

[9] T. Bengtsson, P. Bickel, and B. Li. Curse of dimensionality revisited: the collapse of importance sampling in very large scale systems. *IMS Collections: Probability and Statistics: Essays in Honor of David Freedman*, 2:316–334, 2008.

[10] A. Bennett. *Inverse Modeling of the Ocean and Atmosphere.* Cambridge, 2002.

[11] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer Verlag, 1985.

[12] N. Berglund and B. Gentz. *Noise-induced Phenomena in Slow-fast Dynamical Systems.* Probability and its Applications (New York). Springer-Verlag London Ltd., London, 2006. A sample-paths approach.

[13] J. Bernardo and A. Smith. *Bayesian Theory.* Wiley, 1994.

[14] A. Beskos, D. Crisan, A. Jasra, et al. On the stability of sequential monte carlo methods in high dimensions. *The Annals of Applied Probability*, 24(4):1396–1445, 2014.

[15] A. Beskos, O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):333–382, 2006.

[16] A. Beskos, G. O. Roberts, A. M. Stuart, and J. Voss. MCMC methods for diffusion bridges. *Stochastic Dynamics*, 8(3):319–350, Sep 2008.

[17] P. Bickel, B. Li, and T. Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. *IMS Collections: Pushing the Limits of Contemporary Statistics*, 3:318–329, 2008.

[18] C. Bishop, B. Etherton, and S. Majumdar. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Mon. Wea. Rev.*, 129(3):420–436, 2001.

[19] M. Branicki and A. Majda. Quantifying bayesian filter performance for turbulent dynamical systems through information theory. *Comm. Math. Sci*, 2014.

[20] L. Breiman. Probability, volume 7 of classics in applied mathematics. *Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA*, 1992.

[21] C. E. A. Brett, K. F. Lam, K. J. H. Law, D. S. McCormick, M. R. Scott, and A. M. Stuart. Accuracy and stability of filters for dissipative pdes. *Physica D: Nonlinear Phenomena*, 245(1):34–45, 2013.

[22] S. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998.

[23] A. Bryson and M. Frazier. Smoothing for linear and nonlinear dynamic systems. In *Proceedings Optimum System Sythensis Conference*. US Air Force Tech. Rep. AFB-TDR-63-119, 1963.

[24] A. Carrassi, M. Ghil, A. Trevisan, and F. Uboldi. Data assimilation as a nonlinear dynamical systems problem: Stability and convergence of the prediction-assimilation system. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 18:023112, 2008.

[25] A. Chorin, M. Morzfeld, and X. Tu. Implicit particle filters for data assimilation. *Comm. Appl. Math. Comp. Sc.*, 5:221–240, 2010.

[26] A. Chorin and X. Tu. Interpolation and iteration for nonlinear filters. http://arxiv.org/abs/0910.3241.

[27] A. Chorin and X. Tu. Implicit sampling for particle filters. *Proc. Nat. Acad. Sc.*, 106:17249–17254, 2009.

[28] S. L. Cotter, M. Dashti, J. C. Robinson, and A. M. Stuart. Bayesian inverse problems for functions and applications to fluid mechanics. *Inverse Problems*, 25(11):115008, 2009.

[29] S. L. Cotter, M. Dashti, and A. M. Stuart. Approximation of Bayesian inverse problems for PDEs. *SIAM J. Num. Anal.*, 48(1):322–345, 2010.

[30] S. L. Cotter, M. Dashti, and A. M. Stuart. Variational data assimilation using targetted random walks. *Int. J. Num. Meth. Fluids*, 2011.

[31] S. L. Cotter, G. Roberts, A. M. Stuart, and D. White. MCMC methods for functions: modifying old algorithms to make them faster. *Statistical Science*, 28:424–446, 2013.

[32] P. Courtier, E. Andersson, W. Heckley, D. Vasiljevic, M. Hamrud, A. Hollingsworth, F. Rabier, M. Fisher, and J. Pailleux. The ECMWF implementation of three-dimensional variational assimilation (3d-Var). I: Formulation. *Quart. J. R. Met. Soc.*, 124(550):1783–1807, 1998.

[33] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746, 2002.

[34] M. Dashti, S. Harris, and A. M. Stuart. Besov priors for Bayesian inverse problems. *Inverse Problems and Imaging*, 6:183–200, 2012.

[35] M. Dashti, K. J. H. Law, A. M. Stuart, and J. Voss. MAP estimators and posterior consistency in bayesian nonparametric inverse problems. *Inverse Problems*, 29:095017, 2013.

[36] M. Dashti and A. M. Stuart. Uncertainty quantification and weak approximation of an elliptic inverse problem. *SIAM J. Num. Anal.*, 49:2524–2542, 2011.

[37] P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. In *Annales de l'Institut Henri Poincaré (B) Probability and Statistics*, volume 37, pages 155–194. Elsevier, 2001.

[38] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[39] N. Doucet, A. de Frietas and N. Gordon. *Sequential Monte Carlo in Practice.* Springer-Verlag, 2001.

[40] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter.* Springer, 2006.

[41] K. J. Falconer. *The Geometry of Fractal Sets*, volume 85. Cambridge University Press, 1986.

[42] M. Fisher, Mand Leutbecher and G. Kelly. On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Q.J.Roy.Met.Soc.*, 131(613):3235–3246, 2005.

[43] M. I. Freidlin and A. D. Wentzell. *Random Perturbations of Dynamical Systems*, volume 260 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, New York, 1984.

[44] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis.* CRC press, 2013.

[45] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

[46] S. Ghosal, J. K. Ghosh, and R. V. Ramamoorthi. Consistency issues in Bayesian nonparametrics. In *Asymptotics, Nonparametrics and Time Series: A Tribute*, pages 639–667. Marcel Dekker, 1998.

[47] A. Gibbs and F. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70:419–435, 2002.

[48] C. González-Tokman and B. R. Hunt. Ensemble data assimilation for hyperbolic systems. *PhysicaD*, 243:128–142, 2013.

[49] G. A. Gottwald and A. Majda. A mechanism for catastrophic filter divergence in data assimilation for sparse observation networks. *Nonlinear Processes in Geophysics*, 20(5):705–712, 2013.

[50] G. Grimmett and D. Stirzaker. *Probability and Random Processes.* Oxford University Press, New York, 2001.

[51] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 1983.

[52] A. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter.* Cambridge Univ Pr, 1991.

[53] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[54] K. Hayden, E. Olson, and E. Titi. Discrete data assimilation in the Lorenz and 2d Navier-Stokes equations. *Physica D: Nonlinear Phenomena*, 2011.

[55] V. H. Hoang, K. J. Law, and A. M. Stuart. Determining white noise forcing from eulerian observations in the navier-stokes equation. *Stochastic Partial Differential Equations: Analysis and Computations*, 2(2):233–261, 2014.

[56] I. Hoteit, X. Luo, and D.-T. Pham. Particle Kalman filtering: A nonlinear Bayesian framework for ensemble Kalman filters. *Mon. Wea. Rev.*, 140(2), 2012.

[57] I. Hoteit, D.-T. Pham, G. Triantafyllou, and G. Korres. A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. *Mon. Wea. Rev.*, 136(1), 2008.

[58] P. Houtekamer and H. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Mon. Wea. Rev.*, 129(1):123–137, 2001.

[59] A. R. Humphries and A. M. Stuart. Deterministic and random dynamical systems: theory and numerics. In *Modern methods in scientific computing and applications (Montréal, QC, 2001)*, volume 75 of *NATO Sci. Ser. II Math. Phys. Chem.*, pages 211–254. Kluwer Acad. Publ., Dordrecht, 2002.

[60] K. Ide, M. Courier, M. Ghil, and A. Lorenc. Unified notation for assimilation: Operational, sequential and variational. *J. Met. Soc. Japan*, 75:181–189, 1997.

[61] K. Ide and C. Jones. Special issue on the mathematics of data assimilation. *PhysicaD*, 230:vii–viii, 2007.

[62] M. A. Iglesias, K. J. H. Law, and A. M. Stuart. Evaluation of Gaussian approximations for data assimilation in reservoir models. *Computational Geosciences*, 17:851–885, 2013.

[63] A. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 63. Academic Pr, 1970.

[64] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*, volume 160 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 2005.

[65] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[66] E. Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability.* Cambridge, 2003.

[67] D. T. B. Kelly, K. J. H. Law, and A. M. Stuart. Well-posedness and accuracy of the ensemble Kalman filter in discrete and continuous time. *Nonlinearity*, 27(10):2579, 2014.

[68] P. Lancaster and L. Rodman. *Algebraic Riccati Equations.* Oxford University Press, 1995.

[69] K. J. H. Law, D. Sanz-Alonso, A. Shukla, and A. M. Stuart. Filter accuracy for chaotic dynamical systems: fixed versus adaptive observation operators. http://arxiv.org/abs/1411.3113.

[70] K. J. H. Law, A. Shukla, and A. M. Stuart. Analysis of the 3DVAR filter for the partially observed Lorenz '63 model. *Discrete and Continuous Dynamical Systems A*, 34:1061–1078, 2014.

[71] K. J. H. Law and A. M. Stuart. Evaluating data assimilation algorithms. *Mon. Wea. Rev.*, 140:3757–3782, 2012.

[72] F. Le Gland, V. Monbet, and V.-D. Tran. Large sample asymptotics for the ensemble kalman filter. 2009.

[73] J. Li and D. Xiu. On numerical properties of the ensemble kalman filter for data assimilation. *Computer Methods in Applied Mechanics and Engineering*, 197(43):3574–3583, 2008.

[74] J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer Series in Statistics. Springer, 2001.

[75] A. C. Lorenc. Analysis methods for numerical weather prediction. *Quart. J. R. Met. Soc.*, 112(474):1177–1194, 2000.

[76] A. C. Lorenc, S. P. Ballard, R. S. Bell, N. B. Ingleby, P. L. F. Andrews, D. M. Barker, J. R. Bray, A. M. Clayton, T. Dalby, D. Li, T. J. Payne, and F. W. Saunders. The Met. Office global three-dimensional variational data assimilation scheme. *Quart. J. R. Met. Soc.*, 126(570):2991–3012, 2000.

[77] E. Lorenz. Deterministic nonperiodic flow. *Atmos J Sci*, 20:130–141, 1963.

[78] E. Lorenz. The problem of deducing the climate from the governing equations. *Tellus*, 16(1):1–11, 1964.

[79] E. Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on Predictability*, volume 1, pages 1–18, 1996.

[80] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons, 1968.

[81] A. Majda, D. Giannakis, and I. Horenko. Information theory, model error and predictive skill of stochastic models for complex nonlinear systems. *Physica D*, 241:1735–1752, 2012.

[82] A. Majda and J. Harlim. *Filtering Complex Turbulent Systems.* Cambridge University Press, 2012.

[83] A. Majda, J. Harlim, and B. Gershgorin. Mathematical strategies for filtering turbulent dynamical systems. *Disc. Cont. Dyn. Sys.*, 2010.

[84] A. Majda and X. Wang. *Nonlinear Dynamics and Statistical Theories for Geophysical Flows.* Cambridge, Cambridge, 2006.

[85] J. Mandel, L. Cobb, and J. D. Beezley. On the convergence of the ensemble Kalman filter. *Applications of Mathematics*, 56(6):533–541, 2011.

[86] N. Metropolis, R. Rosenbluth, M. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[87] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability.* Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1993.

[88] A. Moodey, A. Lawless, R. Potthast, and P. van Leeuwen. Nonlinear error dynamics for cycled data assimilation methods. *Inverse Problems*, 29(2):025002, 2013.

[89] L. Nerger, T. Janjic, J. Schröter, and W. Hiller. A unification of ensemble square root Kalman filters. *Mon. Wea. Rev.*, 140(7):2335–2345, 2012.

[90] N. Nichols. Data assimilation: aims and basic concepts. In *Data Assimilation for the Earth System, Editors R. Swinbank, V.Shutyaev, W.A.Lahoz*, pages 9–20. Kluwer, 2003.

[91] J. Nocedal and S. Wright. *Numerical Optimization.* Springer verlag, 1999.

[92] J. Norris. *Markov Chains.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 1998.

[93] D. Oliver, A. Reynolds, and N. Liu. *Inverse Theory for Petroleum Reservoir Characterization and History Matching.* Cambridge Univ Pr, 2008.

[94] E. Ott, B. Hunt, I. Szunyogh, A. Zimin, E. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. Yorke. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A*, 56:273–277, 2004.

[95] D. F. Parrish and J. C. Derber. The national meteorological centers spectral statistical-interpolation analysis system. *Mon. Wea. Rev.*, 120(8):1747–1763, 1992.

[96] D. Pham, J. Verron, and L. Gourdeau. Singular evolutive Kalman filters for data assimilation in oceanography. *Oceanographic Literature Review*, 45(8), 1998.

[97] P. Rebeschini and R. van Handel. Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab. (To appear)*, 2015.

[98] S. Reich. A Gaussian-mixture ensemble transform filter. *Q.J.Roy.Met.Soc.*, 138(662):222–233, 2012.

[99] C. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer Texts in Statistics. Springer-Verlag, 1999.

[100] D. Sanz-Alonso and A. M. Stuart. Long-time asymptotics of the filtering distribution for partially observed chaotic dynamical systems. *arXiv:1406.1936*, 2014.

[101] E. Schröder. Ueber iterirte functionen. *Mathematische Annalen*, 3(2):296–322, 1870.

[102] L. Slivinski, E. Spiller, A. Apte, and B. Sandstede. A hybrid particle-ensemble Kalman filter for lagrangian data assimilation. *Mon. Wea. Rev.*, 145:195–211, 2015.

[103] T. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Mon. Wea. Rev..*, 136:4629–4640, 2008.

[104] E. Sontag. *Mathematical Control Theory.* Springer, 1998.

[105] C. Sparrow. *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors*, volume 41 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 1982.

[106] A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numer.*, 19:451–559, 2010.

[107] A. M. Stuart and A. R. Humphries. *Dynamical Systems and Numerical Analysis*, volume 2 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, Cambridge, 1996.

[108] A. Tarantola. *Inverse Problem Theory.* SIAM, 2005.

[109] R. Temam. *Infinite-Dimensional Dynamical Systems in Mechanics and Physics*, volume 68 of *Applied Mathematical Sciences.* Springer-Verlag, New York, second edition, 1997.

[110] M. Tippett, J. Anderson, C. Bishop, T. Hamill, and J. Whitaker. Ensemble square root filters. *Mon. Wea. Rev.*, 131(7):1485–1490, 2003.

[111] W. Tucker. The Lorenz attractor exists. *C. R. Acad. Sci. Paris Sér. I Math.*, 328(12):1197–1202, 1999.

[112] W. Tucker. A rigorous ODE solver and Smale's 14th problem. *Found. Comput. Math.*, 2(1):53–117, 2002.

[113] A. W. Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge university press, 2000.

[114] P. Van Leeuwen. Particle filtering in geophysical systems. *Mon. Wea. Rev.*, 137:4089–4114, 2009.

[115] P. van Leeuwen. Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Q.J.Roy.Met.Soc.*, 136(653):1991–1999, 2010.

[116] P. Van Leeuwen and G. Evensen. Data assimilation and inverse methods in terms of a probabilistic formulation. *Mon. Wea. Rev.*, 124:2892–2913, 1996.

[117] M. Viana. *Stochastic Dynamics of Deterministic Systems*, volume 21. IMPA Brazil, 1997.

[118] J. Whitaker and T. Hamill. Ensemble data assimilation without perturbed observations. *Mon. Wea. Rev.*, 130(7):1913–1924, 2002.

[119] S. Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, volume 2 of *Texts in Applied Mathematics.* Springer-Verlag, New York, 2003.

[120] D. Williams. *Probability with Martingales.* Cambridge University Press, Cambridge, 1991.

[121] J. Zabczyk. *Mathematical Control Theory: An Introduction.* Springer, 2009.

[122] D. Zupanski. A general weak constraint applicable to operational 4DVAR data assimilation systems. *Mon. Wea. Rev.*, 125:2274–2292, 1997.

Index

σ-algebra, 4
  Borel, 4
3DVAR, 50, 81, 82, 93, 96, 100, 101, 103–106, 108, 109, 132, 134
4DVAR, 54, 65, 70, 71, 73, 82, 116
  strong constraint, 67
  weak constraint, 54, 67, 71, 116, 127

absorbing set, 14, 15, 31, 50
allocation, 116
almost surely, 5
analysis, 37, 38, 77
atmospheric sciences, 32, 49, 65
autocorrelation, 58

background
  covariance, 37
  mean, 35, 37
  penalization, 35, 37, 114
Bayes' formula, 9, 10, 22, 33–35, 37, 77–79, 86, 87, 91, 92
Bayesian, 33, 43–45, 50, 51, 54, 65, 67, 68, 73, 86, 93
Bayesian quality assessment, 44, 45, 93
Borel
  σ-algebra, 4
  set, 4, 5, 14
burn-in, 119
burn-in time, 72, 116

calculus of variations, 65, 67
chaos, 31, 89
conditional distribution, 9
consistency checks, 51
covariance, 78

data, 25, 26, 33, 36, 39, 44, 54, 57, 61
  accumulated, 37, 77
data assimilation, 4, 11, 16, 26, 43, 44, 48, 51, 53, 85
  window, 33
detailed balance, 58–60
deterministic dynamics, 26, 29, 45, 53, 65, 77, 98, 114, 127
diffusion process, 73
Dirac mass, 7, 86
distance, 16, 17, 45
  Hellinger, 18, 19, 24
  total variation, 17, 19, 24
distributed, 5
dynamical system, 11, 15, 32
  continuous-time, 11, 13
  discrete, 14, 15
  discrete-time, 11
  ergodic, 22
  stochastic, 12, 77

empirical measure, 29
equilibrium point, 13, 51
ergodic, 22, 23, 27, 29, 30, 49, 57, 58, 60, 68, 69, 89, 108
  dynamical system, 22
  theorem, 22
event, 5
expected value, 5

filter
  approximate Gaussian, 49, 81–83, 85, 93, 103, 105, 108
  bootstrap, 85, 87, 92
  particle, 38, 50, 53
  synchronization, 98, 109, 132
filtering, 25, 37–39, 42–44, 50, 77, 97, 103
  algorithm, 77, 129, 145
  analysis step, 38, 77–79, 83, 132
  distribution, 39, 42, 50, 52, 77, 78, 85, 87, 89, 91, 109, 110, 129, 138, 140, 142
  prediction step, 38, 78, 81, 83, 86
  program, 129, 136

update, 77
fixed point, 11, 22, 29, 47, 93–95
forecast skill, 44, 45, 51

Galerkin, 109
Gaussian, 6, 26
geophysical sciences, 48–50
subsurface, 50, 51
global attractor, 15, 31, 32, 50, 89
Gronwall, 97
Gronwall lemma, 15
continuous time, 14
discrete time, 11

histogram, 29, 68, 110
rank, 51, 110

i.i.d., 10, 25, 26, 34, 37, 46, 59, 77, 83, 111, 112, 117, 121, 138
identical twin experiments, 51
Independence Dynamics Sampler, 61–64, 68–72, 75, 89, 119, 121, 124
innovation, 80, 130, 138, 140, 142
invariant density, 22, 30
invariant measure, 22, 27, 31, 57

Kalman filter, 49, 77–81, 83, 93, 95, 96, 99, 103, 108, 110, 129, 140
EnKF, 82
ensemble, 50, 81–83, 85, 100–109, 140, 142
ensemble square root, 83
ensemble transform, 84, 140
ESRF, 107
ETKF, 84, 109
ExKF, 82
extended, 50, 81, 82, 100–102, 104–106, 108, 140
square root, 107
Kalman gain, 80, 130, 138, 140
Kalman smoother, 53, 54, 56, 64
krigging, 104
Kullback-Leibler divergence, 24

likelihood, 10, 23, 34, 35, 38, 87, 89, 91, 104
log-likelihood, 62
localization, 107
Lorenz model, 147
’63, 30, 109
’96, 32, 109
LU factorization, 71

Lyapunov function, 23

MAP estimator, 65, 68, 70, 71, 73
marginal distribution, 8, 43
Markov chain, 20, 22, 23, 25, 57–61, 63, 64, 70, 72, 75, 86, 89, 90, 116, 119
Markov inequality, 5
Markov kernel, 20, 63, 77, 89–91
maximum *a posteriori* estimators, 65
MCMC, 54, 57, 59, 60, 62–64, 68, 69, 72, 116
measurable, 20, 23
measure
Lebesgue, 5
metric, 17
Hellinger, 18, 20, 51, 76
total variation, 17, 51, 52, 76, 88
Metropolis-Hastings, 54, 57–61, 76
mixing time, 72
model error, 43–45, 51
model misspecification, 44
model-data misfit, 35, 37, 67, 114
Monte Carlo, 53, 85
approximation, 83
hybrid, 74
Markov Chain, 54, 57, 72
sampling, 53, 85

noise
additive, 25
normalization constant, 10, 48, 54, 79

observation operator, 26, 54
observations, 25, 26, 47, 81, 98, 109
oceanography, 49
oil recovery, 49
Onsager-Machlup functional, 73
optimal interpolation, 104
optimization, 54, 65, 67, 70, 71, 73, 116, 127

particle filter, 77, 85, 86, 88, 89, 91, 92, 100, 101, 108
pCN, 49, 62–64, 68, 70, 72–75, 121, 124
perturbed observations, 82, 83, 101, 105, 108, 138, 140
posterior, 23
log-posterior, 35–37, 61, 62, 64
posterior consistency, 43, 46
Bayesian, 43
posterior distribution, 35, 37, 44, 45, 47, 51, 54, 57, 60, 70, 71, 73, 77, 108, 114, 116, 119, 127

preallocation, 114, 116, 119
precision, 8, 55, 78, 79
prediction, 37, 77
prior, 23, 34
probability
  conditional, 9
  marginal, 8
  triple, 4
probability density function, 4, 24, 29, 49, 58, 67
probability measure, 16, 18, 20, 23, 33, 37
proposal, 60–63, 89, 91, 100, 116, 117, 119, 121, 124, 145
  covariance, 60, 61
  Gaussian, 60, 62, 70
  independence dynamics sampler, 61
  index, 108
  optimal, 77, 92, 101, 106, 108
  pCN, 62, 64
  pCN dynamics sampler, 63
  RWM, 60, 62
  standard, 92, 101, 105
  variance, 60, 64, 124
pushforward, 5, 36, 38, 39, 89

Radon-Nikodym derivative, 86
random variable, 4
  Gaussian, 6
  normal, 6
Random Walk Metropolis, 60, 62, 68, 76, 116
  RWM, 60
reanalysis, 51
RWM, 60

seed, 113, 115, 127
semigroup, 13, 23
sequential importance resampling, 86
shadowing, 109
signal, 25, 26, 33, 39, 44, 51, 54, 57, 61
signal estimation quality assessment, 44, 45, 93
smoothing, 25, 37–39, 41–44, 47, 53
  distribution, 33, 35–37, 39, 114, 116, 119
solution operator, 13, 26, 44
stochastic differential equations, 49
stochastic dynamics, 25, 26, 28, 35, 36, 39–45, 52–55, 57, 61–65, 67–69, 75–77, 112, 113, 119
synchronization, 98, 109, 132

truth, 43, 44, 47, 95

variance inflation, 97–99, 105–107
variational method, 54, 64, 65, 68, 70, 116

weak convergence, 5, 57
weather forecasting, 44, 49–51
well-posed, 25
Woodbury matrix identity, 80