(b) Perform the same analysis using linear models that incorporate only x_1 as well as x_1 and x_2. How do your results compare with those obtained in (a)?

Obs. No.	x_1	x_2	x_3	x_4	v
1	7	26	6	60	78.5
2	1	29	15	52	74.3
3	11	56	8	20	104.3
4	11	31	8	47	87.6
5	7	52	6	33	95.9
6	11	55	9	22	109.2
7	3	71	17	6	102.7
8	1	31	22	44	72.5
9	2	54	18	22	93.1
10	21	47	4	26	115.9
11	1	40	23	34	83.8
12	11	66	9	12	113.3
13	10	68	8	12	109.4

Table 7.4. *Cement data from* [104].

Chapter 8

Bayesian Techniques for Parameter Estimation

For applications where modeling and measurement errors ε_i are unbiased and iid, we employ the statistical model

$$\Upsilon_i = f_i(Q) + \varepsilon_i \ , \ i = 1, \ldots, n, \tag{8.1}$$

where $\Upsilon_i, \varepsilon_i$, and Q are random variables representing measurements, measurement errors, and parameters. As defined in (7.9), $f_i(Q)$ denotes the parameter-dependent model response. We note that the measurement errors in this case are modeled as additive and mutually independent from Q. We also remind readers that calibration parameters and observed data are commonly denoted by θ and y in the statistics literature.

8.1 Parameter Estimation from a Bayesian Perspective

As detailed in Section 4.8, the tenets of Bayesian inference differ significantly from the frequentist perspective described in Chapter 7. In the context of inverse problems involving parameter estimation, the Bayesian approach can be summarized as follows. Parameters are considered to be random variables Q with realizations $q = Q(\omega)$ and associated densities that incorporate known information or information obtained as measurements are acquired. The solution of the inverse problem is the posterior density that best reflects the distribution of parameter values based on the sampled observations.

It was shown in Section 4.8.2 that the posterior is constructed in terms of a prior density and likelihood. The prior density $\pi_0(q)$ incorporates any knowledge that we have about parameters prior to obtaining observations v. This could come from previous similar experiments or analysis regarding similar models. It was illustrated that if prior knowledge is of questionable accuracy, it is better to use a noninformative prior which is often taken as an improper uniform density posed on the parameter support; for example, one would employ $\pi_0(q) = \chi_{(0,\infty)}(q)$ for positive parameters.

The likelihood function $\pi(\upsilon|q) = L(q|\upsilon)$ incorporates information provided by the samples and constitutes the mechanism through which data informs the posterior density. As detailed in Section 4.3.2, the likelihood quantifies the probability of obtaining the observations υ for a given value q of the parameter Q. Hence if we let $\pi(q,\upsilon)$ denote the joint density of Q and Υ, then the likelihood

$$\pi(\upsilon|q) = \frac{\pi(q,\upsilon)}{\pi_0(q)}$$

is the conditional probability of Υ given a value of Q.

Once we have a measurement or observation $\upsilon = \upsilon_{obs}$, the conditional density

$$\pi(q|\upsilon_{obs}) = \frac{\pi(q,\upsilon_{obs})}{\pi(\upsilon_{obs})},$$

where we assume that

$$\pi(\upsilon_{obs}) = \int_{\mathbb{R}^p} \pi(q,\upsilon_{obs})dq = \int_{\mathbb{R}^p} \pi(\upsilon_{obs}|q)\pi_0(q)dq \neq 0,$$

is the posterior density. The inverse problem in the Bayesian framework can thus be stated as follows: given measurements υ_{obs}, find the posterior density $\pi(q|\upsilon_{obs})$. The complete formulation, which the authors of [128] refer to as *Bayes' theorem of inverse problems,* can be stated as follows.

Result 8.1 (Bayes' Theorem of Inverse Problems). We assume that the p random parameter variables Q have a known prior density $\pi_0(q)$, which can be noninformative, and we let υ_{obs} be a realization of the random observation variable Υ. The posterior density of Q, given the measurements υ_{obs}, is

$$\pi(q|\upsilon_{obs}) = \frac{\pi(\upsilon_{obs}|q)\pi_0(q)}{\pi(\upsilon_{obs})} = \frac{\pi(\upsilon_{obs}|q)\pi_0(q)}{\int_{\mathbb{R}^p} \pi(\upsilon_{obs}|q)\pi_0(q)dq}. \tag{8.2}$$

When using (8.2), one implicitly assumes that observed data is used to construct the posterior density; hence we write $\upsilon = \upsilon_{obs}$ in subsequent discussion so that (8.2) is the same as (4.41).

8.1.1 Likelihood Function

The specification of the likelihood function $\pi(\upsilon|q)$ depends on the assumptions made regarding the distribution of errors. In Section 4.3.2, we showed that if we employ the statistical model (8.1) with the assumption that errors are iid and $\varepsilon_i \sim N(0,\sigma^2)$, where σ^2 is fixed, then the likelihood function is

$$\pi(\upsilon|q) = L(q,\sigma^2|\upsilon) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-SS_q/2\sigma^2}, \tag{8.3}$$

where

$$SS_q = \sum_{i=1}^{n} [\upsilon_i - f_i(q)]^2 \tag{8.4}$$

is the sum of squares error. The construction of likelihoods for other error models, including multiplicative noise, is addressed in [128].

8.1.2 Maximum a Posteriori (MAP) Estimate

The posterior density $\pi(q|\upsilon)$ provides the complete distribution of Q based on the observations υ. From this, point estimates for the parameter values are provided by the mean, median, or mode. The latter is defined as the parameter value that maximizes $\pi(q|\upsilon)$. This value, termed the MAP estimate, is given by

$$q_{MAP} = \operatorname*{argmax}_q \pi(q|\upsilon).$$

Since the normalization constant $\pi(\upsilon)$ does not affect the maximizing argument, an equivalent formulation is

$$q_{MAP} = \operatorname*{argmax}_q \pi(\upsilon|q)\pi_0(q). \tag{8.5}$$

For a uniform prior on $\mathbb{R}$, q_{MAP} is thus equivalent to the maximum likelihood estimate q_{MLE} defined in (4.28). As detailed in Section 4.3.2, one would typically employ the log-likelihood function $\ell(q,\sigma|\upsilon)$ in such cases since it facilitates optimization by eliminating the exponential.

8.1.3 Implementation Techniques

The formulation of the inverse problem in the Bayesian framework is concisely provided by Result 8.1. However, the implementation of (8.2) is extremely challenging if the dimensionality p of Q is large, as is often the case for physical or biological models. As illustrated in the next example, classical tensored quadrature rules can be applied for low dimensionality; e.g., $p \leq 6$. Within the last ten years, significant research has focused on the development of adaptive sparse grid quadrature techniques for moderate dimensionality and Monte Carlo techniques for high dimensions. These techniques are discussed in Chapter 11.

Alternatively, one can construct Markov chains whose stationary distribution is the posterior density. We discuss Markov chain Monte Carlo (MCMC) techniques in Section 8.2.

Example 8.2. Consider the spring model

$$\ddot{z} + C\dot{z} + Kz = 0,$$
$$z(0) = 2\ ,\ \dot{z}(0) = -C,$$

which, for $C^2 - 4K < 0$, has the solution

$$z(t) = 2e^{-Ct/2}\cos(\sqrt{K - C^2/4}\cdot t).$$

We assume displacement evaluation so that $y(t_i,Q) = z(t_i,Q)$. We consider $K = 20.5$ to be known and treat $Q = C$ as the unknown parameter to be estimated. To construct synthetic data, we take $C_0 = 1.5$ and construct iid errors $\varepsilon_i \sim N(0,\sigma_0^2)$, where $\sigma_0 = 0.1$.

For this error distribution, the likelihood is given by (8.3). We employ the noninformative prior $\pi_0(q) = \chi_{[0,\infty)}(q)$ to enforce C to be nonnegative. The posterior density can thus be expressed as

$$\pi(q|\upsilon) = \frac{e^{-SS_q/2\sigma_0^2}}{\int_0^\infty e^{-SS_\zeta/2\sigma_0^2}d\zeta} = \frac{1}{\int_0^\infty e^{-(SS_\zeta - SS_q)/2\sigma_0^2}d\zeta},$$

where SS_q is defined in (8.4) and SS_ζ denotes the sum of squares defined in terms of the integration variable. The second formulation is necessary to avoid numerical $\frac{0}{0}$ evaluation since $e^{-SS_{q_{MAP}}} \approx 3\times 10^{-113}$. The use of a midpoint rule to approximate the integral yields

$$\pi(q|\upsilon) \approx \frac{1}{\sum_{i=1}^k e^{-(SS_{\zeta^i} - SS_q)/2\sigma_0^2}w^i}, \tag{8.6}$$

where ζ^i, w^i respectively denote the quadrature points and weights.

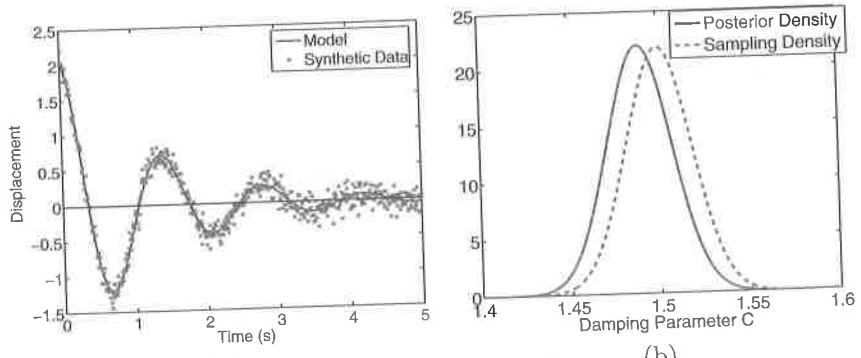
We generated one set of synthetic data υ_i, $i = 1, \ldots, 501$, which is plotted in Figure 8.1(a) along with the model response $f_i(q_0)$. The posterior density given by (8.6) is plotted in Figure 8.1(b). We note that the MAP estimate is $q_{MAP} = 0.1489$. Since we have employed a noninformative prior, this corresponds to the MLE. For the assumed error distribution, it also corresponds to the OLS estimate.

We showed in Chapter 7 that the OLS estimator has the sampling distribution

$$\hat{q}_{OLS} = \hat{C}_{OLS} \sim N\left(C_0, \sigma_0^2[\mathcal{X}^T(C_0)\mathcal{X}(C_0)]^{-1}\right), \tag{8.7}$$

where $\mathcal{X}(C_0)$ is given in (7.43) of Example 7.15. The sampling distribution is compared with the posterior density in Figure 8.1(b). We note that they have the same shape but the sampling distribution is centered at C_0.

We will revisit this example in Example 8.7, where we construct $\pi(q|\upsilon)$ using MCMC methods.



(a) (b)

Figure 8.1. (a) *Synthetic data* υ_i *and model response* $f_i(q_0)$. (b) *Posterior density and sampling distribution* (8.7).

8.2 Markov Chain Monte Carlo (MCMC) Techniques

The evaluation of the posterior relation (8.2) using quadrature techniques requires the evaluation of densities over the region of $\mathbb{R}^p$ where the posterior is defined. For moderate p, this necessitates the use of the sparse grid quadrature techniques discussed in Chapter 11, whereas Monte Carlo integration techniques are required for large dimensionality p. The difficulty of this approach is exacerbated by the fact that the support of the density is often part of the information that we are seeking.

An alternative is the following. Rather than using quadrature or Monte Carlo algorithms to specify parameter values at which we evaluate the density, we can use attributes of the density to specify parameter values that adequately explore the geometry of the distribution. This is achieved by constructing Markov chains whose stationary distribution, as defined in Definition 4.52, is the posterior density. By evaluating realizations of the chain, one thus samples the posterior and hence obtains a density for the parameter values based on observed measurements. This is the basis for the MCMC techniques employed here.

In Section 8.3, we summarize the Metropolis and Metropolis–Hastings algorithms and motivate their structure. The detailed balance condition defined in Definition 4.62 is used in Section 8.4 to establish that $\pi(q|\upsilon)$ is the stationary distribution for the chain. We also discuss convergence criteria in that section. The role of parameter identifiability is discussed in Section 8.5, and the development of the delayed rejection adaptive Metropolis (DRAM) algorithm is detailed in Section 8.6. This is the algorithm that we employ in subsequent chapters. The DiffeRential Evolution Adaptive Metropolis (DREAM) algorithm is summarized in Section 8.7. The reader is referred to Section 4.6 for relevant definitions and theory pertaining to Markov chains.

8.3 Metropolis and Metropolis–Hastings Algorithms

Recall from Definition 4.50 that a Markov chain is a sequence of S-valued random variables that satisfy the Markov property that X_k depends only on X_{k-1}. The state space in this case is the set of possible parameter values, so we will be constructing chains based on parameters chosen according to the following strategy.

Strategy 8.3. Consider the parameter $q^{k-1} \in \mathbb{R}^p$ to be specified.

(i) Take the current chain realization to be $X_{k-1} = q^{k-1}$.

(ii) Propose a new value $q^* \sim J(q^*|q^{k-1})$, where J is called the proposal or jumping distribution. The notation indicates that J specifies q^* based on the previous value q^{k-1}, and $J(q^*|q^{k-1})$ should not be interpreted as a conditional density.

(iii) With probability $\alpha(q^*|q^{k-1})$, determined by properties of the likelihood function and prior density, accept q^*; i.e., $X_k = q^*$. Otherwise, take $X_k = q^{k-1}$. We note that $\alpha(q^*|q^{k-1})$ is not a conditional probability but rather specifies the probability of accepting q^* generated from the previous value q^{k-1}.

(iv) Establish that the posterior density is the stationary distribution for the chain.

8.3.1 Metropolis Algorithm

We consider first the case when the proposal distribution is taken to be symmetric in the sense that $J(q^*|q^{k-1}) = J(q^{k-1}|q^*)$. We consider two possibilities for the proposal distribution:

$$J(q^*|q^{k-1}) = N(q^{k-1}, V),$$
$$J(q^*|q^{k-1}) = N(q^{k-1}, D). \tag{8.8}$$

Here V is the covariance matrix for Q, whereas D is a diagonal matrix whose elements reflect the scale associated with each parameter value. The symmetry in the first case follows since

$$\begin{aligned} J(q^*|q^{k-1}) &= \frac{1}{\sqrt{(2\pi)^p|V|}} e^{-\frac{1}{2}[(q^*-q^{k-1})V^{-1}(q^*-q^{k-1})^T]} \\ &= \frac{1}{\sqrt{(2\pi)^p|V|}} e^{-\frac{1}{2}[(q^{k-1}-q^*)V^{-1}(q^{k-1}-q^*)^T]} \\ &= J(q^{k-1}|q^*). \end{aligned}$$

The analysis of the second choice is similar. We will provide further motivation for these choices after we summarize the algorithm.

Algorithm 8.4 (Metropolis Algorithm).

1. Initialization: Choose an initial parameter value q^0 that satisfies $\pi(q^0|v) > 0$.
2. For $k = 1, \ldots, M$
   (a) For $z \sim N(0,1)$, construct the candidate

   $$q^* = q^{k-1} + Rz,$$

   where R is the Cholesky decomposition of V or D. As specified in Theorem 4.23, this ensures that

   $$q^* \sim N(q^{k-1}, V) \text{ or } q^* \sim N(q^{k-1}, D).$$

   Because the construction of q^* takes into account q^{k-1}, this is termed a *random walk* or *local Metropolis algorithm.*

   (b) Compute the ratio

   $$r(q^*|q^{k-1}) = \frac{\pi(q^*|v)}{\pi(q^{k-1}|v)} = \frac{\pi(v|q^*)\pi_0(q^*)}{\pi(v|q^{k-1})\pi_0(q^{k-1})}. \tag{8.9}$$

   (c) Set

   $$q^k = \begin{cases} q^* & , \text{ with probability } \alpha = \min(1, r), \\ q^{k-1} & , \text{ else.} \end{cases}$$

   That is, we accept q^* with probability 1 if $r \geq 1$ and we accept it with probability r if $r < 1$.

We first motivate the choice of acceptance criteria in steps 2(b) and (c). The first observation is that by forming the ratio of the posterior densities, we eliminate the normalization constant, which is difficult to compute when p is moderate or large. Now consider the case of a uniform prior and iid and normally distributed errors so that the likelihood is

$$\pi(v|q) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-SS_q/2\sigma^2} \;,\; SS_q = \sum_{i=1}^{n} [v_i - f_i(q)]^2, \tag{8.10}$$

as established in (8.3) and (8.4). With these assumptions

$$r(q^*|q^{k-1}) = \frac{\pi(v|q^*)}{\pi(v|q^{k-1})} = \frac{e^{-SS_{q^*}/2\sigma^2}}{e^{-SS_{q^{k-1}}/2\sigma^2}} = e^{-[SS_{q^*} - SS_{q^{k-1}}]/2\sigma^2}, \tag{8.11}$$

where the final step eliminates the potential for numerical $\frac{0}{0}$ evaluation, as noted in Example 8.2. As illustrated in Figure 8.2, a candidate q^* that yields $\pi(v|q^*) > \pi(v|q^{k-1})$ is equivalent to producing a smaller sum of squares error and this candidate is accepted with probability one. If q^* is such that $\pi(v|q^*) < \pi(v|q^{k-1})$, and hence the sum of squares error is increased, we accept the candidate with probability $\alpha = r$.

Properties of the proposal function and how they affect mixing are illustrated in Figures 8.3 and 8.4. If the variance is too large, a large percentage of the candidates will be rejected since they will have smaller likelihoods, and hence the chain will stagnate for long periods. The acceptance ratio will be high if the variance is small, but the algorithm will be slow to explore the parameter space.

As illustrated in Figure 8.4(a), if the posterior is highly anisotropic but the proposal distribution is isotropic, the efficiency with which the algorithm explores with respect to various components of the parameter vector will be highly nonuniform. The choices (8.8) for $J(q^*|q^{k-1})$ address these issues by scaling the variability of each parameter component in the manner depicted in Figure 8.4(b). The goal is to achieve, to the degree possible, the efficiency of the univariate case.

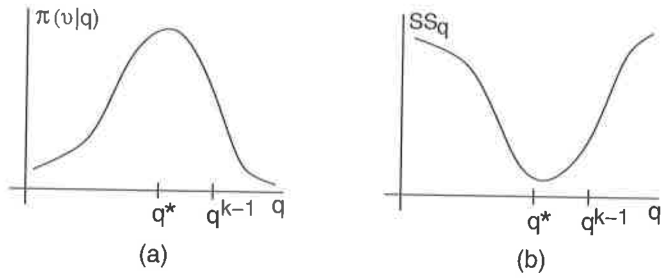


Figure 8.2. (a) *Likelihood and* (b) *sum of squares functions of* Q.

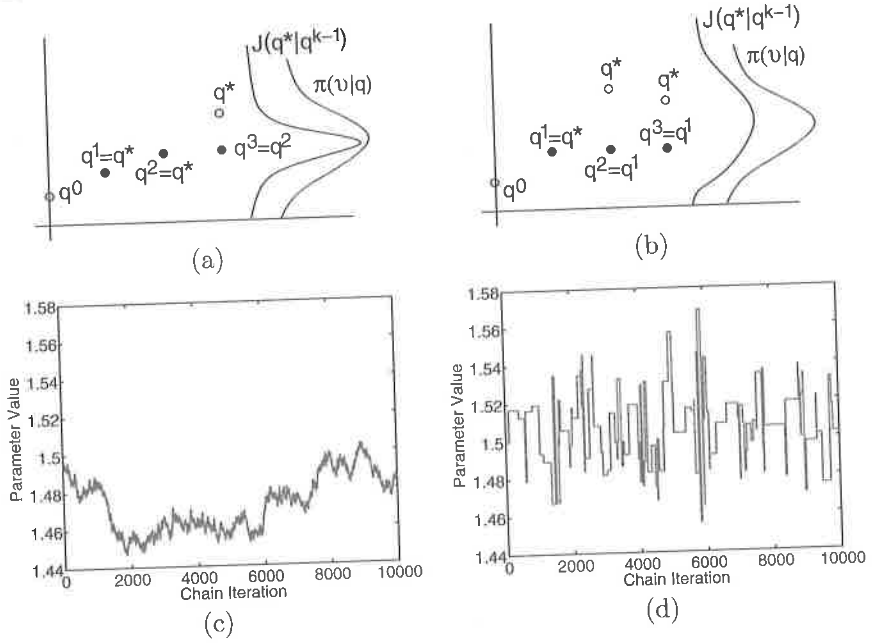


Figure 8.3. *Generation of candidates q^* based on* (a) *narrow and* (b) *wide proposal functions $J(q^*|q^{k-1})$. Chains resulting from proposal functions that are* (c) *too narrow and* (d) *too wide.*

The covariance matrix V is estimated in the manner detailed in Section 7.3. Specifically, we take

$$\begin{aligned} V &= \sigma_{OLS}^2 \left[\mathcal{X}^T(q_{OLS})\mathcal{X}(q_{OLS})\right]^{-1}, \\ \sigma_{OLS}^2 &= \frac{1}{n-p}\sum_{i=1}^{n}\left[\upsilon_i - f_i(q_{OLS})\right]^2, \end{aligned} \tag{8.12}$$

where $\mathcal{X}_{ik}(q) = \frac{\partial f_i(q)}{dq_k}$, as indicated in Table 7.3 on page 146.

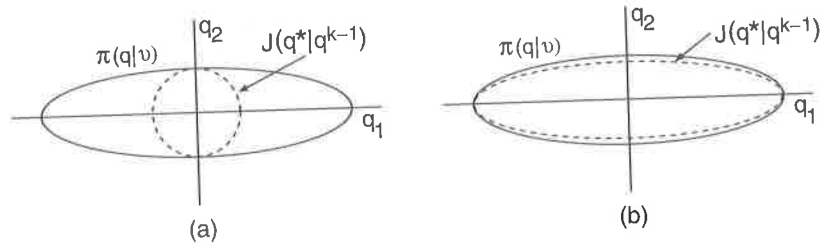


Figure 8.4. *Anisotropic posterior $\pi(q|\upsilon)$ and* (a) *isotropic and* (b) *anisotropic proposal functions $J(q^*|q^{k-1})$.*

8.3.2 Sample-Based Error Variance

The assumption that errors are iid and $\varepsilon_i \sim N(0,\sigma^2)$ yields the likelihood function (8.10) and acceptance ratio (8.11) formulated in terms of σ^2. In most applications, however, σ^2 is fixed but unknown. One solution is to employ the estimate (8.12) for σ^2. Alternatively one can treat it as an additional random parameter whose density is sampled through realizations of the Markov chain.

As illustrated in Example 4.69, the likelihood

$$\pi(\upsilon, q|\sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}}e^{-SS_q/2\sigma^2}$$

is in the inverse-gamma family, detailed in Definition 4.14, so the conjugate prior is

$$\pi_0(\sigma^2) \propto (\sigma^2)^{-(\alpha+1)}e^{\beta/\sigma^2}. \tag{8.13}$$

The hyperparameters α and β can be treated as design parameters. The resulting posterior density representation is

$$\pi(\sigma^2|q,\upsilon) \propto (\sigma^2)^{-(\alpha+1+n/2)}e^{-(\beta+SS_q/2)/\sigma^2}$$

so that

$$\sigma^2|(\upsilon,q) \sim \text{Inv-gamma}\left(\alpha+\frac{n}{2}, \beta+\frac{SS_q}{2}\right). \tag{8.14}$$

An equivalent representation is

$$\sigma^2|(\upsilon,q) \sim \text{Inv-gamma}\left(\frac{n_s+n}{2}, \frac{n_s\sigma_s^2+SS_q}{2}\right), \tag{8.15}$$

where $n_s = 2\alpha$ and $\sigma_s^2 = \frac{\beta}{\alpha}$. As noted in [92], n_s can be interpreted as representing the number of observations that provided the information encoded in the prior, whereas σ_s^2 represents the mean squared error of the observations. In practice, one often takes n_s to be small (e.g., $n_s = 0.01$ to 1), which is consistent with a noninformative prior.

As noted in Example 4.69 and Definitions 4.13 and 4.14, random numbers from an inverse gamma distribution can be generated using the MATLAB Statistics Toolbox command `gamrnd.m` and then exploiting the equivalence between the gamma and inverse gamma distributions. If `gamrnd.m` is not available, one can use the inverse transform techniques of Section 4.1.1 to generate samples from the inverse gamma distribution.

We summarize in Algorithm 8.5 the random walk Metropolis algorithm with sampling-based error variance and noninformative prior. Some implementations employ the OLS estimate σ_{OLS}^2 from (8.12) or the previous estimate s_{k-1}^2 for σ_s^2. Whereas this is in the spirit of "empirical Bayes" inference as discussed in Section 4.8.2, it is noted in [35] that use of the present data to inform the prior can be problematic with small sample sizes and is at odds with the tenets of Bayesian analysis.

The issues associated with subjectively specifying n_s and σ_s^2 based on prior knowledge can be avoided by using the Jeffreys prior

$$\pi_0(q, \sigma^2) = \frac{1}{\sigma^2}.$$

It is illustrated in Section 3.3.3 of [34] that this relation results from specification of a prior based on the Fisher information matrix for problems with mutually independent location and scale parameters q and σ^2. Alternatively, it can be obtained from (8.13) in the limit $\alpha \to 0$, $\beta \to 0$ for the hyperparameters.

Algorithm 8.5 (Random Walk Metropolis with Noninformative Prior).

```
1. Set number of chain elements $M$ and design parameters $n_s, \sigma_s$
2. Determine $q^0 = \arg\min_q \sum_{i=1}^n [\upsilon_i - f_i(q)]^2$
3. Set $SS_{q^0} = \sum_{i=1}^n [\upsilon_i - f_i(q^0)]^2$
4. Compute initial variance estimate: $s_0^2 = \frac{SS_{q^0}}{n-p}$
5. Construct covariance estimate $V = s_0^2[\mathcal{X}^T(q^0)\mathcal{X}(q^0)]^{-1}$ and $R = \text{chol}(V)$
6. For $k = 1, \ldots, M$
   (a) Sample $z_k \sim N(0, I_p)$
   (b) Construct candidate $q^* = q^{k-1} + Rz_k$
   (c) Sample $u_\alpha \sim \mathcal{U}(0,1)$
   (d) Compute $SS_{q^*} = \sum_{i=1}^n [\upsilon_i - f_i(q^*)]^2$
   (e) Compute
          $\alpha(q^*|q^{k-1}) = \min\left(1, e^{-[SS_{q^*} - SS_{q^{k-1}}]/2s_{k-1}^2}\right)$
   (f) If $u_\alpha < \alpha$,
          Set $q^k = q^*$ , $SS_{q^k} = SS_{q^*}$
       else
          Set $q^k = q^{k-1}$ , $SS_{q^k} = SS_{q^{k-1}}$
       endif
   (g) Update $s_k^2 \sim \text{Inv-gamma}(a_{val}, b_{val})$, where
          $a_{val} = 0.5(n_s + n)$ , $b_{val} = 0.5(n_s\sigma_s^2 + SS_{q^k})$
```

Remark 8.6. We noted in (7.32) that for models in which the parameter scales vary by several orders of magnitude, one typically employs the scaled parameter $q_s = q./s$ in optimization routines. Here s is a vector whose elements are the magnitude of each parameter and ./ denotes componentwise division. The same scaling can improve the efficiency of optimization routines used to determine q^0, the conditioning of V, and the efficiency of Algorithm 8.5. Specifically, one would employ the alternative steps:

2. Determine $q_s^0 = \arg\min_{q_s} \sum_{i=1}^n [\upsilon_i - f_i(q_s. \times s)]^2$ and $q^0 = q_s^0. \times s$.
5. Construct covariance estimate $V = s_0^2[\mathcal{X}^T(q_s^0. \times s)\mathcal{X}^T(q_s^0. \times s)]^{-1}$ and $R = \text{chol}(V)$.
6. (b) Construct candidate $q_s^* = q_s^{k-1} + Rz_k$, and set $q^* = q_s^*. \times s$.
6. (f) Additionally set $q_s^k = q_s^*$ or $q_s^k = q_s^{k-1}$.

Note that the unscaled parameters q are employed in all model evaluations $f_i(q)$.

8.3.3 Metropolis–Hastings Algorithm

The Metropolis–Hastings algorithm generalizes the Metropolis algorithm to include nonsymmetric jumping or proposal functions $J(q^*|q^{k-1})$. For example, this includes Cauchy distributions

$$J(q^*|q^{k-1}) = \frac{1}{\pi[1 + (q^*)^2]}$$

and $\chi^2(k)$ distributions

$$J(q^*|q^{k-1}) = \kappa(q^*)^{k/2-1}e^{q^*/2}.$$

In this case, candidates are accepted with probability $\alpha = \min(1, r)$, where the acceptance ratio is

$$\begin{aligned} r(q^*|q^{k-1}) &= \frac{\pi(q^*|\upsilon)/J(q^*|q^{k-1})}{\pi(q^{k-1}|\upsilon)/J(q^{k-1}|q^*)} \\ &= \frac{\pi(\upsilon|q^*)\pi_0(q^*)J(q^{k-1}|q^*)}{\pi(\upsilon|q^{k-1})\pi_0(q^{k-1})J(q^*|q^{k-1})}. \end{aligned} \tag{8.16}$$

For symmetric proposal functions $J(q^*|q^{k-1}) = J(q^{k-1}|q^*)$, (8.16) reduces to (8.9). We focus primarily on the Metropolis algorithm with symmetric proposal functions and refer the reader to [92] for details regarding the Metropolis–Hastings algorithm.

Example 8.7. In Examples 7.15 and 8.2, we considered the estimation of the parameter C for the spring model

$$\ddot{z} + C\dot{z} + Kz = 0,$$

$$z(0) = 2\ ,\ \dot{z}(0) = -C$$

from a frequentist perspective and direct implementation of Bayes' relation (8.2). Here we illustrate the random walk Metropolis Algorithm 8.5. Recall that for $C^2 - 4K < 0$, the solution is

$$z(t) = 2e^{-Ct/2}\cos(\sqrt{K - C^2/4} \cdot t).$$

We consider displacement measurements at $n = 501$ points in the time interval $[0, 5]$ so that $y_i(Q) = z(t_i, Q)$, where $t_i = 0.01i$. Synthetic data v_i is simulated with errors $\varepsilon_i \sim N(0, \sigma_0^2)$, where $\sigma_0 = 0.1$ is considered unknown when implementing the MCMC algorithm.

Case i. To compare with Example 8.2, we first take $K = 20.5$ to be known and generate the displacement response with $C_0 = 1.5$. Hence the random parameters are $Q = [C, \sigma^2]$. We consider chains of length $M = 10,000$.

The chain, or marginal path, and kernel density estimate for C, computed using `kde.m`, are plotted in Figure 8.5. The comparison between the density computed using the random walk Metropolis algorithm and the direct posterior evaluation detailed in Example 8.2 shows that the two are nearly identical. The MCMC kernel will converge in the sense of distributions as M is increased and quadrature errors when computing the normalization constant are decreased. We note that the marginal path for C provides a baseline for comparison when applying the algorithm for multiple model parameters.

Case ii. Second, we consider the estimation of densities for $Q = [C, K, \sigma^2]$ using synthetic data generated with $K_0 = 20.5$, $C_0 = 1.5$, and $\sigma_0 = 0.1$. We consider first the choice $J(q^*|q^{k-1}) = N(q^{k-1}, V)$ for the proposal distribution. Here

$$V = \begin{bmatrix} 0.000345 & 0.000268 \\ 0.000268 & 0.007071 \end{bmatrix}$$

is the covariance matrix given by (8.12) which is constructed using the analytic sensitivity relations

$$\frac{\partial y}{\partial C} = e^{-Ct/2}\left[\frac{Ct}{\sqrt{4K - C^2}} \sin\left(\sqrt{K - C^2/4} \cdot t\right) - t\cos\left(\sqrt{K - C^2/4} \cdot t\right)\right],$$

$$\frac{\partial y}{\partial K} = \frac{-2t}{\sqrt{4K - C^2}} e^{-Ct/2} \sin\left(\sqrt{K - C^2/4} \cdot t\right).$$

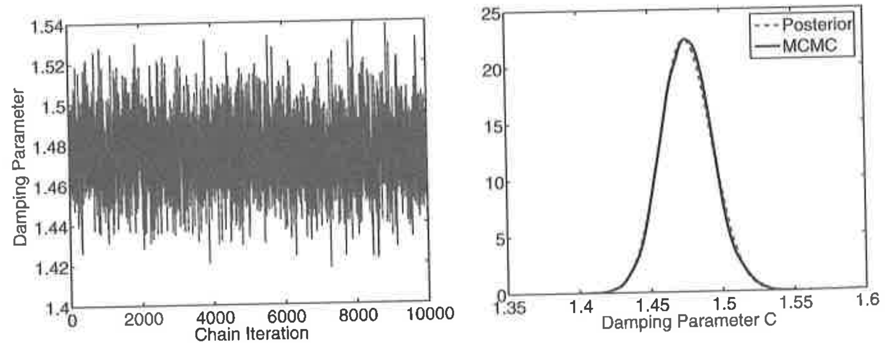


Figure 8.5. *Marginal path and density for the damping parameter C.*

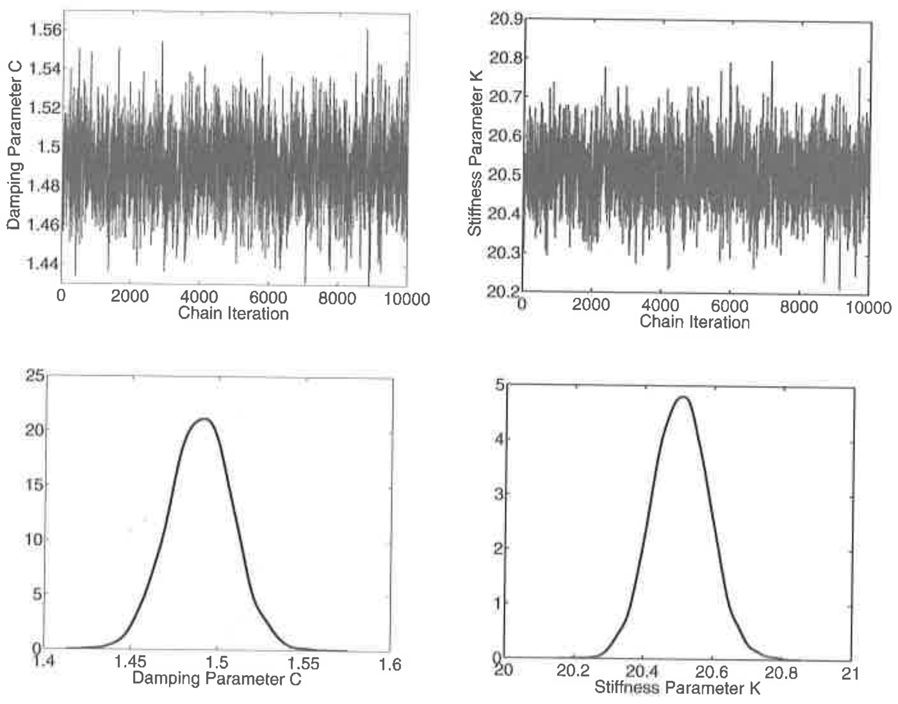
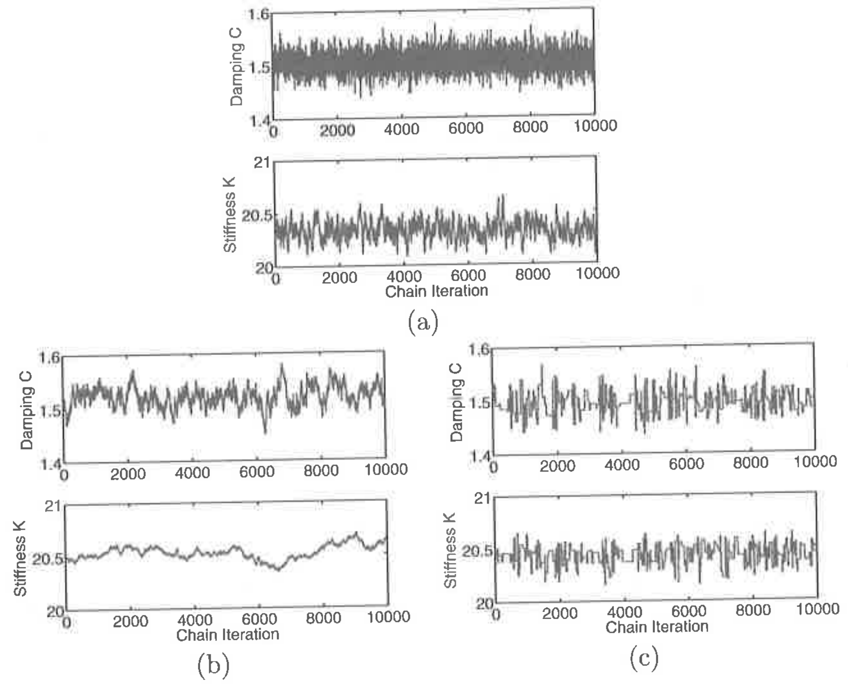


Figure 8.6. *Marginal paths and densities for the damping parameter C and stiffness parameter K obtained with $J(q^*|q^{k-1}) = N(q^{k-1}, V)$.*

The marginal paths and densities are plotted in Figure 8.6. It is observed that $2\sigma_C \approx 0.04$ so that $\sigma_C^2 \approx 0.4 \times 10^{-3}$, whereas $2\sigma_K \approx 0.18$ so that $\sigma_K^2 \approx 0.0081$. These are close to the variance values in the covariance matrix V, which illustrates how it incorporates the general anisotropy exhibited by the posterior density. As a result, the marginal paths in Figure 8.6 exhibit essentially the same degree of mixing as the previous case $Q = [C, \sigma^2]$ plotted in Figure 8.5(a).

To contrast, we illustrate the results obtained with the isotropic proposal function $J(q^*|q^{k-1}) = N(q^{k-1}, sI)$ in Figure 8.7 for three choices of s. The mixing in Figure 8.7(a), which was obtained with $s = 9 \times 10^{-4}$, is reasonable but is not as rich as that obtained with the anisotropic proposal function $J(q^*|q^{k-1}) = N(q^{k-1}, V)$. Figure 8.7(b) illustrates the results obtained using a narrower proposal function constructed with $s = 9 \times 10^{-6}$. This yields substantial mixing but poor exploration of the parameter space for K. Conversely, the choice 9×10^{-2} yields poor mixing and chain stagnation since a large number of candidates are rejected. This illustrates the advantage of using the covariance matrix when it can be accurately constructed. Alternatively, in Section 8.6, we will discuss delayed rejection adaptive Metropolis methods that can be used to update the proposal distribution as candidates are accepted and the geometry of the posterior is determined.



(a)

(b)

(c)

Figure 8.7. *Sample paths obtained with the proposal functions* $J(q^*|q^{k-1}) = N(q^{k-1}, sI)$*:* (a) $s = 9 \times 10^{-4}$, (b) $s = 9 \times 10^{-6}$, *and* (c) $s = 9 \times 10^{-2}$.

8.4 Stationary Distribution and Convergence Criteria

The random walk Metropolis algorithm provides a Markov chain whose state space is the set of admissible parameter values. The initial distribution is provided by an OLS fit to calibration data. However, it is important to note that the Markov chain is based on samples from the "wrong" distribution in the sense that it is constructed using the proposal density rather than the sought after posterior density. In this section, we address two questions: (i) why should we expect the chain to have a stationary distribution that coincides with the posterior density, and (ii) what criteria indicate that the chain has converged to this distribution?

In Definition 4.62, we showed that the detailed balance condition $\pi_{k-1}p_{k-1,k} = \pi_k p_{k,k-1}$ was a sufficient (but not necessary) requirement for stationarity. Since we want to show that the posterior density is the stationary distribution, we take $\pi_k = \pi(q^k|\upsilon)$. Similarly, we consider

$$p_{k-1,k} = P(X_k = q^k | X_{k-1} = q^{k-1}),$$

which is the probability of transitioning from parameter q^{k-1} to q^k. The detailed balance condition in this context can thus be expressed as

$$\begin{aligned} &\pi_{k-1}p_{k-1,k} = \pi_k p_{k,k-1} \\ \Rightarrow\ &\pi(q^{k-1}|\upsilon)p_{k-1,k} = \pi(q^k|\upsilon)p_{k,k-1}. \end{aligned}$$

Since $p_{k-1,k} = P(\text{proposing } q^k)P(\text{accepting } q^k)$, it follows from the definition of the proposal distribution $J(q^k|q^{k-1})$ and acceptance probability α that

$$\begin{aligned} p_{k-1,k} &= J(q^k|q^{k-1})\alpha(q^k|q^{k-1}) \\ &= J(q^k|q^{k-1}) \min\left(1, \frac{\pi(q^k|\upsilon)J(q^{k-1}|q^k)}{\pi(q^{k-1}|\upsilon)J(q^k|q^{k-1})}\right). \end{aligned}$$

From the relation

$$\upsilon \min(1, x/\upsilon) = \min(x, \upsilon) = x \min(1, \upsilon/x),$$

which is established in Exercise 8.1, it follows that

$$\begin{aligned} \pi(q^{k-1}|\upsilon)p_{k-1,k} &= \pi(q^{k-1}|\upsilon)J(q^k|q^{k-1}) \min\left(1, \tfrac{\pi(q^k|\upsilon)J(q^{k-1}|q^k)}{\pi(q^{k-1}|\upsilon)J(q^k|q^{k-1})}\right) \\ &= \pi(q^k|\upsilon)J(q^{k-1}|q^k) \min\left(1, \tfrac{\pi(q^{k-1}|\upsilon)J(q^k|q^{k-1})}{\pi(q^k|\upsilon)J(q^{k-1}|q^k)}\right) \qquad (8.17) \\ &= \pi(q^k|\upsilon)p_{k,k-1}. \end{aligned}$$

Hence the detailed balance condition is satisfied for the Metropolis–Hastings acceptance relation and the posterior density is the stationary distribution. We note that the transition kernel for the Markov chain can be defined as

$$\begin{aligned} p_{ij} &= J(q^j|q^i) \min\left(1, \frac{\pi(q^j|\upsilon)J(q^i|q^j)}{\pi(q^i|\upsilon)J(q^j|q^i)}\right), \quad i \neq j, \\ p_{ii} &= 1 - \sum_{j \neq i} p_{ij}. \end{aligned}$$

The detailed balance result (8.17) establishes that if chains are run sufficiently long, they will produce samples from the posterior density. However, the question of how long chains must be run to converge to and adequately sample from the posterior is difficult and analytic convergence and stopping criteria are lacking. It is noted in [42] that the convergence, or burn-in, of MCMC algorithms can be falsified but, in general, not completely verified.

Despite the lack of analytic convergence theory, there are various tests that can be used to establish confidence in simulations. We summarize only aspects of these tests and refer readers to [42, 92] for details regarding convergence or burn-in of MCMC simulations.

The most direct method for assessing burn-in or convergence is to visually or statistically monitor the marginal paths associated with each parameter, as illustrated in Figures 8.5, 8.6, and 8.7. The initial period during which means appear to transition is often termed the *burn-in period*, and these values are excluded when computing parameter or response densities since they are not sampled from the stationary or posterior distribution.

The difficulty is that chains can appear stationary for a very large number of simulations and then change in the manner shown in Figure 8.8 for a parameter from a transductive material model [116]. Because MCMC algorithms will determine

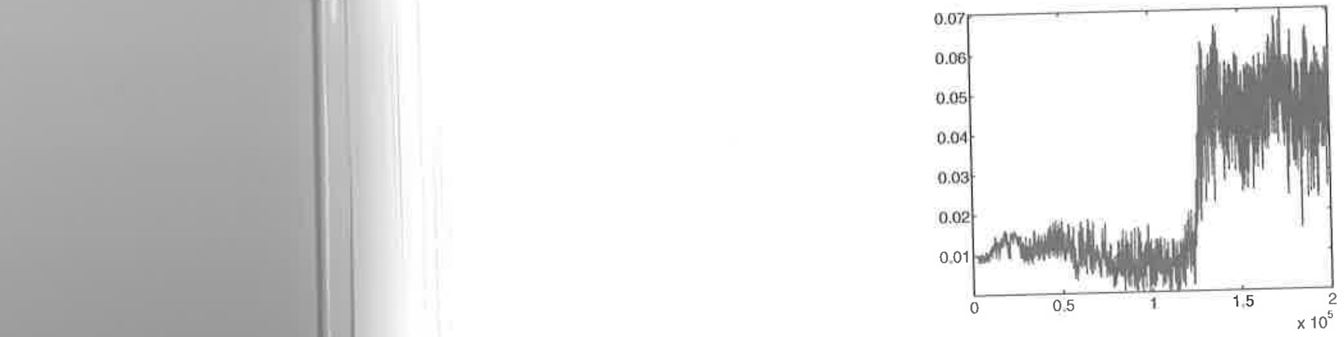


Figure 8.8. *Shift in the marginal path after* 130,000 *iterations due to a local minimum in the sum of squares; see* [116].

global minima, if run sufficiently long, this can be due to initial sampling in a local minimum before determining another with a lower residual. However, there is no guarantee that this is a global minimum and the chain could transition again if another, lower, minimum is found.

In some cases, the parameter density constructed using the burned-in MCMC chain can be compared with that directly computed using Bayes' relation. Whereas this is feasible only for a moderate number of parameters, which may require adaptive sparse grid quadrature, it can be used to verify (or falsify) the MCMC results.

From a statistical perspective, the percentage of accepted points, termed the *acceptance ratio*, is often used to quantify whether or not the chain is adequately sampling from the posterior. Because the optimal acceptance ratio depends on the geometry of the posterior, the range of reasonable acceptance ratios is quite large; e.g., values between 0.1 and 0.5 are often considered acceptable. The acceptance ratio is often used to tune the proposal density $J(q^*|q^{k-1})$ to improve mixing. For example, a small acceptance ratio can produce stagnation, as shown in Figures 8.3(d) and 8.7(c). This can be addressed by decreasing the variance of affected parameters to narrow the proposal function.

A second commonly employed statistical test is to check the autocorrelation

$$R(k) = \frac{\sum_{i=1}^{M-k}(q_i - \bar{q})(q_{i+k} - \bar{q})}{\sum_{i=1}^{M}(q_i - \bar{q})^2} = \frac{\text{cov}(q_i, q_{i+k})}{\text{var}(q_i)} \tag{8.18}$$

between components in the chain that are k iterations apart. Because adjacent components are likely correlated due to the Markov property, this test can be used to establish that the chain is producing iid samples from the posterior. As detailed in [56], low autocorrelation is often indicative of fast convergence.

The reader is warned that care must be exercised when interpreting MCMC results reported in the literature. One commonly encounters parameter densities with no mention of the burn-in period or illustration of marginal sample paths. In such cases, it is difficult to verify that the reported density is truly indicative of the posterior. Second, it is not uncommon for authors employing computationally intensive codes to discuss very short burn-in periods. While this may be the case, it can also reflect the fact that the reported length reflects the largest number of simulations that could be run with the code.

8.5 Parameter Identifiability

We noted in Definition 6.1 that the concept of parameter identifiability quantifies the uniqueness of the input-output map between parameters and responses. *Hence parameter identifiability is a property of the model and observations rather than of the inference or estimation procedure.* For example, we illustrated in Example 3.2 that we could not uniquely determine $q = [m, c, k]$ in the spring model (3.5) given displacement measurements $z(t)$. Instead we had to reformulate the model in terms of the parameters $K = \frac{k}{m}$ and $C = \frac{c}{m}$. As detailed in Chapter 6, one must reformulate the model or fix certain parameter values to address lack of parameter identifiability.

From the perspective of the likelihood, unidentifiability produces flat regions in the likelihood function or multiple maxima having the same value, as illustrated in Figure 8.9(a). From a Bayesian perspective, we noted in Section 6.3 that unidentifiability can be manifested as posterior joint densities that are nearly single-valued for parameters having independent priors. Figure 8.9(b) illustrates the correlation exhibited by unidentifiable material parameters ε_R^{90} and ε_R^{+} in the model of [116]. The fact that multiple parameter values yield the same maximum likelihood value can also cause chains to jump in the manner shown in Figure 8.8. For noninformative priors, this can slow or stop the convergence of the chains to the posterior density. As detailed in Section 6.3, however, it is often difficult to differentiate between identifiable and unidentifiable parameters based solely on the width of joint densities, so this criterion should be interpreted as merely an indicator that parameter values may not be uniquely determined by the data.

In the random walk Metropolis algorithm, it follows from Property 6.7 that lack of identifiability is manifested by a singular covariance matrix V constructed

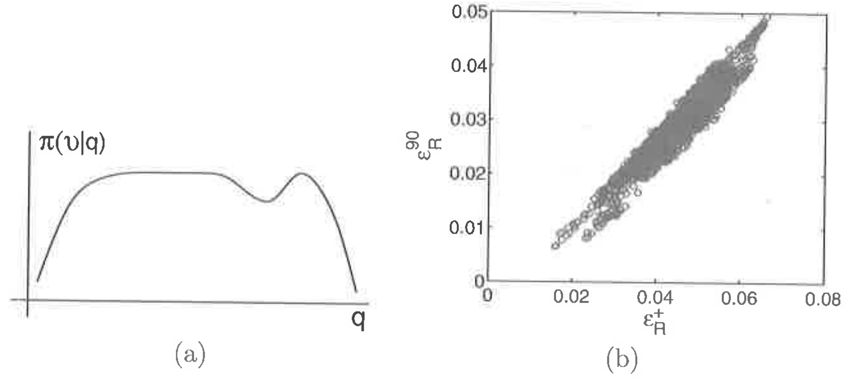


Figure 8.9. (a) *Likelihood for an unidentifiable parameter set and* (b) *correlation of unidentifiable parameters; see* [116].

from the sensitivity relations $\mathcal{X}(q)$. This is illustrated for the simple harmonic oscillator model in Exercise 8.5. This reinforces the relation between V and the Fisher information matrix $\mathcal{F}$ which quantifies the information content of an experiment.

Whereas unidentifiable parameters cannot be uniquely determined using OLS estimators or maximum likelihood estimators, they can in some cases be determined using Bayesian estimators with informative priors for the unidentifiable parameters. Hence Bayesian inference can sometimes be successful for overparameterized or unidentifiable models if informative priors are available.

8.6 Delayed Rejection Adaptive Metropolis (DRAM)

Whereas the choices (8.8) for the proposal distribution incorporate aspects of parameter scaling and variability, they do not provide mechanisms to incorporate information learned about the posterior distribution as candidate parameters are accepted and the chain progresses. Such mechanisms are provided by various adaptive Metropolis algorithms [11, 103, 208, 255], including the DRAM algorithm [102], which we summarize here.

We note that because adaptive algorithms employ part of the chain history to update the proposal function, they are no longer Markovian processes, which requires that states depend only on the previous state. Hence the convergence criteria discussed in Section 8.4 do not apply and alternative ergodicity properties must be established to guarantee convergence to the posterior density. General criteria, such as the *diminishing adaptation* and *bounded convergence conditions*, that adaptive methods must satisfy to establish convergence to a stationary distribution are provided in [11, 103, 208].

8.6.1 Adaptive Metropolis

In principle, the adaptive Metropolis (AM) step employed in the DRAM algorithms is quite straightforward. During a nonadaptive period of length k_0, chain values $q^0, q^1, \ldots, q^{k-1}$ are computed using the initial covariance matrix $V_0 = V$ or $V_0 = D$ employed in the random walk Metropolis Algorithm 8.5. Once adaptation commences, the updated chain covariance matrix at the k^{th} step is taken to be

$$V_k = s_p \mathrm{cov}(q^0, q^1, \ldots, q^{k-1}) + \varepsilon I_p. \tag{8.19}$$

Here s_p is a design parameter that depends on the dimension p of the parameter space. As detailed in [102], a common choice is $s_p = 2.38^2/p$. The length k_0 of the adaptation interval is chosen to balance mixing with providing sufficient diversity in points to ensure a nonsingular covariance matrix in the initial stages of the chain progression. Shorter adaptation intervals typically produce better mixing and higher acceptance ratios since they accelerate the rate at which information regarding the posterior is incorporated. In practice, k_0 is often specified to be approximately 100. The term εI_p, where $\varepsilon \geq 0$ and I_p is the p-dimensional identify matrix, ensures that V_k is positive definite. One can often take $\varepsilon = 0$.

In theory, $\mathrm{cov}(q^0, \ldots, q^{k-1})$ can be computed using the empirical covariance formula

$$\mathrm{cov}(q^0, \ldots, q^{k-1}) = \frac{1}{k-1}\left(\sum_{i=0}^{k-1} q^i (q^i)^T - k\bar{q}^k(\bar{q}^k)^T\right),$$

where $\bar{q}^k = \frac{1}{k}\sum_{i=0}^{k-1} q^i$ and q^i are column vectors. However, this becomes increasingly inefficient as k becomes large. Instead, one employs the recursive relation

$$V_{k+1} = \frac{k-1}{k}V_k + \frac{s_p}{k}\left[k\bar{q}^{k-1}(\bar{q}^{k-1})^T - (k+1)\bar{q}^k(\bar{q}^k)^T + q^k(q^k)^T + \varepsilon I_p\right]. \tag{8.20}$$

In a similar manner, the sample mean can be computed recursively as

$$\begin{aligned}\bar{q}^{k+1} &= \frac{1}{k+1}\sum_{i=0}^{k} q^i \\ &= \frac{k}{k+1}\cdot\frac{1}{k}\sum_{i=0}^{k-1} q^i + \frac{1}{k+1}q^k \\ &= \frac{k}{k+1}\bar{q}^k + \frac{1}{k+1}q^k.\end{aligned}$$

It is noted in [102] that the efficiency of the algorithm is improved if adaptation occurs at prescribed intervals and, in Algorithm 8.8, we employ intervals of length k_0. The ergodicity of this adaptive algorithm is established in [103].

8.6.2 Delayed Rejection

In the standard Metropolis algorithm, chain candidates q^* are accepted with probability

$$\alpha(q^*|q^{k-1}) = \min\left(1, \frac{\pi(q^*|v)J(q^{k-1}|q^*)}{\pi(q^{q-1}|v)J(q^*|q^{k-1})}\right) = \min\left(1, \frac{\pi(q^*|v)}{\pi(q^{k-1}|v)}\right)$$

and, if rejected, the prior chain value q^{k-1} is retained. The delayed rejection (DR) algorithm provides a mechanism for constructing alternative candidates q^{*j} if q^* is rejected rather than initially retaining the previous value.

As detailed in [102], a second-stage candidate q^{*2} is chosen using the proposal function

$$J_2(q^{*2}|q^{k-1}, q^*) = N(q^{k-1}, \gamma_2^2 V_k),$$

where $V_k = R_k R_k^T$ is the covariance matrix produced by the adaptive algorithm. The notation $J_2(q^{*2}|q^{k-1}, q^*)$ indicates that we are proposing q^{*2} having started at q^{k-1} and rejected q^*. The software discussed in Remark 8.9 employs $\gamma_2 = \frac{1}{5}$, but other values are reasonable. Because $\gamma_2 < 1$, the second-stage proposal function is narrower than the original, which increases mixing. The probability of accepting the second-stage candidate, having started at q^{k-1} and rejected q^*, is

$$\begin{aligned}\alpha_2(q^{*2}|q^{k-1},q^*) &= \min\left(1, \frac{\pi(q^{*2}|\upsilon)J(q^*|q^{*2})J_2(q^{k-1}|q^{*2},q^*)[1-\alpha(q^*|q^{*2})]}{\pi(q^{k-1}|\upsilon)J(q^*|q^{k-1})J_2(q^{*2}|q^{k-1},q^*)[1-\alpha(q^*|q^{k-1})]}\right)\\ &= \min\left(1, \frac{\pi(q^{*2}|\upsilon)J(q^*|q^{*2})[1-\alpha(q^*|q^{*2})]}{\pi(q^{k-1}|\upsilon)J(q^*|q^{k-1})[1-\alpha(q^*|q^{k-1})]}\right)\end{aligned} \tag{8.21}$$

due to the symmetry of J_2.

The form of α_2 can be motivated as follows. It was noted in Section 8.4 that for the Metropolis–Hastings algorithm,

$$\begin{aligned}p_{k-1,k} &= P(X_k = q^k|X_{k-1} = q^{k-1})\\ &= P(\text{proposing } q^k)P(\text{accepting } q^k)\\ &= J(q^k|q^{k-1})\alpha(q^k|q^{k-1}).\end{aligned}$$

We now consider the case when we accept $q^k = q^{*2}$ having rejected q^* so that

$$\begin{aligned}p_{k-1,k} &= P(\text{proposing } q^*)P(\text{rejecting } q^*)P(\text{proposing } q^k)P(\text{accepting } q^k)\\ &= J(q^*|q^{k-1})[1-\alpha(q^*|q^{k-1})]J_2(q^k|q^{k-1},q^*)\alpha_2(q^k|q^{k-1},q^*).\end{aligned}$$

To satisfy the detailed balance condition $\pi(q^{k-1}|\upsilon)p_{k-1,k} = \pi(q^k|\upsilon)p_{k,k-1}$, we thus require

$$\begin{aligned}&\pi(q^{k-1}|\upsilon)J(q^*|q^{k-1})[1-\alpha(q^*|q^{k-1})]J_2(q^k|q^{k-1},q^*)\alpha_2(q^k|q^{k-1},q^*)\\ &\quad = \pi(q^k|\upsilon)J(q^*|q^k)[1-\alpha(q^*|q^k)]J_2(q^{k-1}|q^k,q^*)\alpha_2(q^{k-1}|q^k,q^*).\end{aligned}$$

The condition (8.21) guarantees that the detailed balance condition is satisfied and that $\alpha \leq 1$.

If q^{*2} is rejected, a third-stage candidate and acceptance condition can be constructed, and recursive relations to construct j^{th}-stage candidates q^{*j} and probabilities $\alpha_i(q^{*j},\ldots,q^{*2},q^*,q^{k-1})$ are provided in [102]. We employ only a second-stage candidate in Algorithm 8.8 since this is the default in the referenced software.

In combination, DR and AM provide two different but complementary mechanisms to modify the proposal function. The AM provides feedback in the sense that information learned about the posterior through accepted chain candidates is used to update the proposal via the chain covariance matrix. The DR is an open loop mechanism that alters the proposal function in a predetermined manner to improve mixing. The modifications from DR are temporary and have the goal of stimulating mixing, whereas the AM mechanism enacts permanent changes that reflect information learned about the posterior. In Algorithm 8.8, we summarize the DRAM algorithm implemented in the referenced software with a second-stage rejection mechanism. The reader is referred to [102] for details and other combinations of the DR and AM components.

Algorithm 8.8 (Delayed Rejection Adaptive Metropolis Algorithm with Noninformative Prior [102]).

```
1. Set design parameters $n_s, \sigma_s^2, k_0$ and number of chain iterates $M$
2. Determine $q^0 = \arg\min_q \sum_{i=1}^n [\upsilon_i - f_i(q)]^2$
3. Set $SS_{q^0} = \sum_{i=1}^n [\upsilon_i - f_i(q^0)]^2$
4. Compute initial variance estimate: $s_0^2 = \frac{SS_{q^0}}{n-p}$
5. Construct covariance estimate $V = s_0^2[\mathcal{X}^T(q^0)\mathcal{X}(q^0)]^{-1}$
     and $R = \text{chol}(V)$
6. For $k = 1, \ldots, M$
     (a) Sample $z_k \sim N(0, I_p)$
     (b) Construct candidate $q^* = q^{k-1} + Rz_k$
     (c) Sample $u_\alpha \sim \mathcal{U}(0,1)$
     (d) Compute $SS_{q^*} = \sum_{i=1}^n [\upsilon_i - f_i(q^*)]^2$
     (e) Compute
             $\alpha(q^*|q^{k-1}) = \min\left(1, e^{-[SS_{q^*} - SS_{q^{k-1}}]/2s_{k-1}^2}\right)$
     (f) If $u_\alpha < \alpha$,
             Set $q^k = q^*$, $SS_{q^k} = SS_{q^*}$
         else
             Enter DR Algorithm 8.10
         endif
     (g) Update $s_k^2 \sim \text{Inv-gamma}(a_{val}, b_{val})$, where
             $a_{val} = 0.5(n_s + n)$ , $b_{val} = 0.5(n_s\sigma_s^2 + SS_{q^k})$
     (h) if $\text{mod}(k, k_0) = 1$
             Update $V_k = s_p\text{cov}\left(q^0, q^1, \ldots, q^k\right)$
         else
             $V_k = V_{k-1}$
         endif
     (i) Update $R_k = \text{chol}(V_k)$
```

Remark 8.9. MATLAB software for Algorithm 8.8 of [102] is available at the websites https://wiki.helsinki.fi/display/inverse/Adaptive+MCMC and http://helios.fmi.fi/~lainema/mcmc/.

Algorithm 8.10 (Delayed Rejection Component of DRAM with Noninformative Prior).

```
1. Set the design parameter $\gamma_2 = \frac{1}{5}$
2. Sample $z_k \sim N(0, I_p)$
3. Construct second-stage candidate $q^{*2} = q^{k-1} + \gamma_2 R_k z_k$
4. Sample $u_\alpha \sim \mathcal{U}(0,1)$
5. Compute $SS_{q^{*2}} = \sum_{i=1}^n [\upsilon_i - f_i(q^{*2})]^2$
6. Compute
           $\alpha_2(q^{*2}|q^{k-1}, q^*)$ using (8.21)
7. If $u_\alpha < \alpha$,
           Set $q^k = q^{*2}$, $SS_{q^k} = SS_{q^{*2}}$
     else
           Set $q^k = q^{k-1}$, $SS_{q^k} = SS_{q^{k-1}}$
     endif
```

Remark 8.11. It was noted in Remark 8.6 that the performance of the algorithm can be significantly enhanced by using scaled parameters $q_s = q./s$ if physical parameter values vary significantly. This can be implemented with the modified steps:

2. Determine $q_s^0 = \arg\min_{q_s} \sum_{i=1}^n [\upsilon_i - f_i(q_s. \times s)]^2$ and $q^0 = q_s^0. \times s$.
5. Construct covariance estimate $V = s_0^2[\mathcal{X}^T(q_s^0. \times s)\mathcal{X}^T(q_s^0. \times s)]$ and $R = \text{chol}(V)$.
6. (b) Construct candidate $q_s^* = q_s^{k-1} + Rz_k$, and set $q^* = q_s^*. \times s$.
6. (f) Additionally set $q_s^k = q_s^*$.

DR 3. Construct second-stage candidate $q_s^{*2} = q_s^{k-1} + \gamma_2 R_k$, and set $q^{*2} = q_s^{*2}. \times s$.

DR 7. Additionally set $q_s^k = q_s^{*2}$ or $q_s^k = q_s^{k-1}$.

Example 8.12. In Example 7.16, we used frequentist analysis to construct sampling distributions for the parameters $q = [\Phi, h]$ in the model

$$\begin{aligned} \frac{d^2T_s}{dx^2} &= \frac{2(a+b)}{ab}\frac{h}{k}[T_s(x) - T_{amb}], \\ \frac{dT_s}{dx}(0) &= \frac{\Phi}{k} \quad , \quad \frac{dT_s}{dx}(L) = \frac{h}{k}[T_{amb} - T_s(L)] \end{aligned} \tag{8.22}$$

for steady state heat conduction in an uninsulated aluminum rod of length L with a source heat flux Φ at $x = 0$. As detailed in Example 3.5, T_s is the steady state temperature, h is a convective heat transfer coefficient, and $k = 2.37$ W·cm^{-1}·°C^{-1} is the thermal conductivity coefficient for aluminum.

Here we construct densities for Φ and h using both the random walk Metropolis Algorithm 8.5 and the delayed rejection adaptive Metropolis Algorithm 8.8. The residual plot in Figure 7.3(b) motivates the assumption that errors are iid and unbiased. We further assume that they are normally distributed with fixed but unknown variance σ_0^2. With these assumptions, we can employ the likelihood relation (8.10). We employ the covariance estimate

$$V_0 = \begin{bmatrix} 2.1034 \times 10^{-2} & -2.0286 \times 10^{-6} \\ -2.0286 \times 10^{-6} & 2.0972 \times 10^{-10} \end{bmatrix} \tag{8.23}$$

of (7.46) as the initial proposal function.

We first employ the nonadaptive algorithm to provide a baseline to illustrate advantages of the DRAM algorithm. The marginal paths, obtained using the random walk Metropolis algorithm with $M = 10^4$ and $M = 10^5$ Monte Carlo iterations, are plotted in Figure 8.10. Because V_0 incorporates the anisotropy due to the dif-

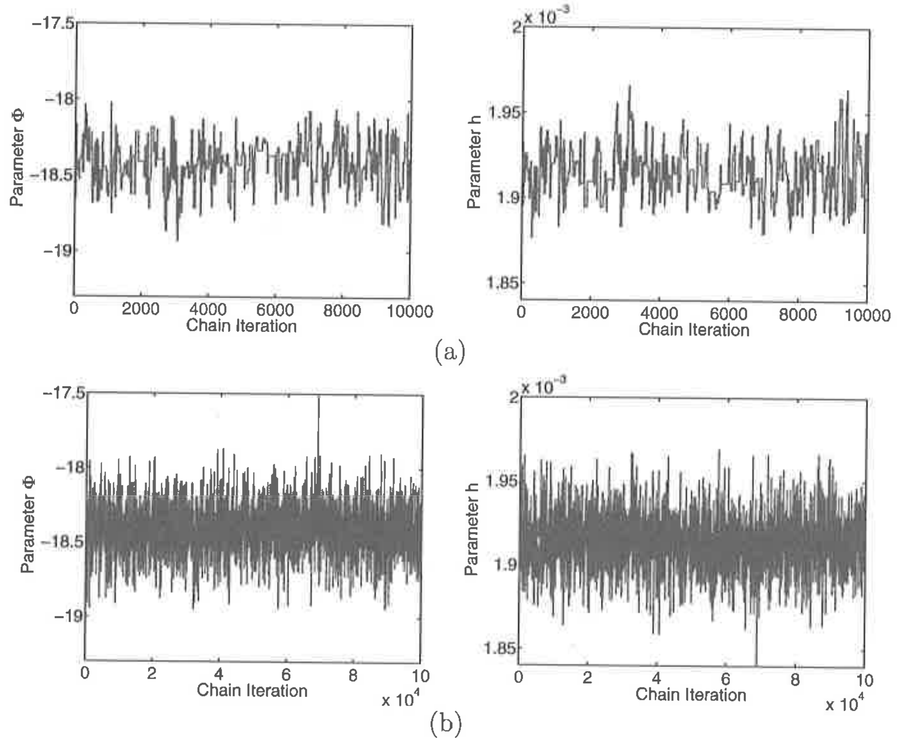


Figure 8.10. *Sample paths obtained using the nonadaptive random walk Metropolis Algorithm 8.5 with* (a) $M = 10^4$ *and* (b) *and* $M = 10^5$ *iterations.*

fering variances of Φ and h, the two chains exhibit similar mixing. However, the acceptance ratio is 0.056, which is smaller than the targeted range $0.1 - 0.5$, and the plots obtained with $M = 10^4$ iterations exhibit regions where the chains briefly stagnate. As illustrated in Figure 8.3, this indicates that narrower proposal functions should improve mixing. We note that the stagnation regions are not visible in the plot of $M = 10^5$ iterates, thus motivating the necessity of checking the acceptance ratio, which remains 0.056. The stationarity of the chains indicates that they have burned-in and are sampling from the posterior density.

The marginal paths and kernel density estimates constructed using $M = 10^4$ DRAM iterates are plotted in Figure 8.11. A comparison with the chains plotted in Figure 8.10(a) and (b) illustrates that the algorithm is achieving the goal of enhancing mixing and accelerating burn-in. The chain covariance matrix is

$$V = \begin{bmatrix} 2.4101 \times 10^{-2} & -2.3211 \times 10^{-6} \\ -2.3211 \times 10^{-6} & 2.3869 \times 10^{-10} \end{bmatrix},$$

which is very close to the original covariance matrix V_0 in (8.23) which was provided by OLS theory. This demonstrates that for this problem, the narrowing of the proposal function in the DR step has a more substantial impact than the proposal modifications in the AM step. The algorithm provides the estimate $\sigma^2 = 0.0678$ for the error variance, so $\sigma = 0.2604$. The estimated standard deviations for Φ and h are $\sigma_\Phi = 0.1552$ and $\sigma_h = 1.5450 \times 10^{-5}$. It is observed in the marginal density plots in Figure 8.11 and joint density plotted in Figure 8.12 that two standard deviations represent approximately 95% of the density.

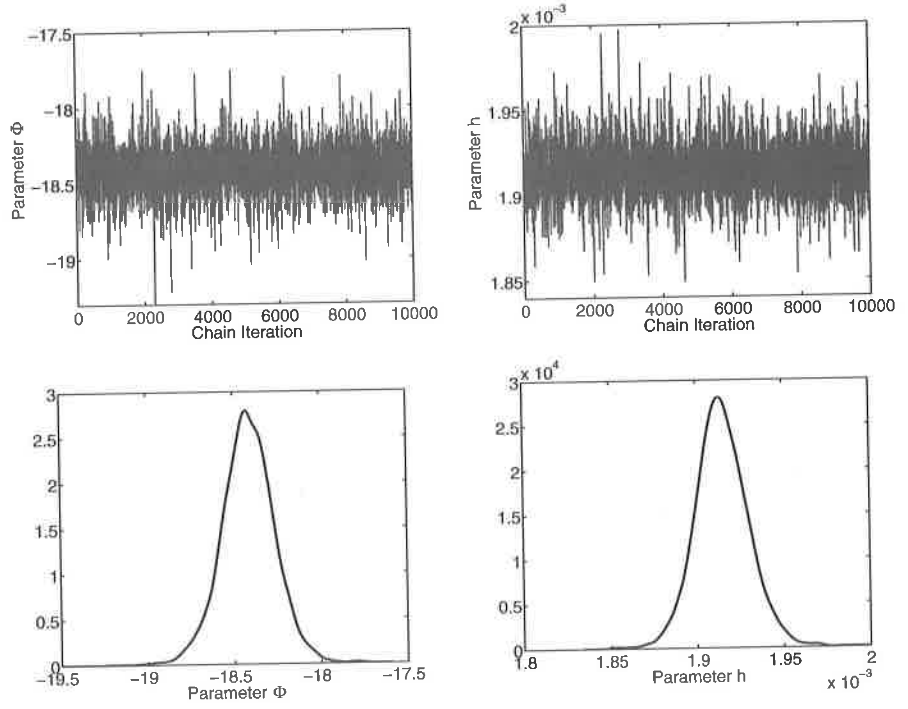


Figure 8.11. *Marginal paths and densities for the parameters Φ and h obtained with the DRAM algorithm.*

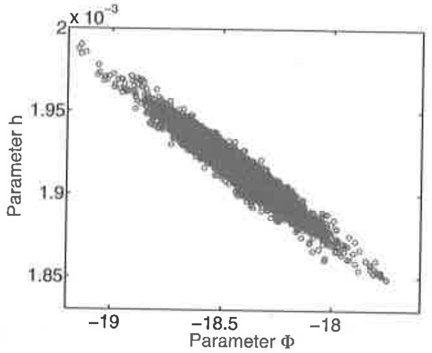


Figure 8.12. *Joint sample points for Φ and h.*

The frequentist analysis in Example 7.16 yielded the standard deviations $\sigma = 0.2504$, $\sigma_\Phi = 0.1450$, and $\sigma_h = 1.4482 \times 10^{-5}$. We first note that the values of σ are within 4% despite the fact that $\sigma = 0.2504$ is an estimate, whereas $\sigma = 0.2604$ is the mean of a density sampled through realizations of the Markov chain. Furthermore, we observe that the values of σ_Φ and σ_h obtained through Bayesian analysis are within 5% of the frequentist values for the sampling distribution.

From a mathematical perspective, the similarity of the sampling distribution and parameter distribution can be attributed to the normality of the estimated parameter densities. However, care must be exercised when interpreting this result since the sampling distribution is for the parameter estimator rather than the parameters. As detailed in Chapter 7, it thus quantifies uncertainty pertaining to the estimation procedure rather than uncertainty associated with the parameters.

Example 8.13. To illustrate the performance of the delayed rejection adaptive Metropolis algorithm for a system of coupled ODEs with multiple responses, we employ the model

$$\begin{aligned}
\dot{T}_1 &= \lambda_1 - d_1 T_1 - (1-\varepsilon) k_1 V T_1, \\
\dot{T}_2 &= \lambda_2 - d_2 T_2 - (1-f\varepsilon) k_2 V T_2, \\
\dot{T}_1^* &= (1-\varepsilon) k_1 V T_1 - \delta T_1^* - m_1 E T_1^*, \\
\dot{T}_2^* &= (1-f\varepsilon) k_2 V T_2 - \delta T_2^* - m_2 E T_2^*, \qquad (8.24) \\
\dot{V} &= N_T \delta (T_1^* + T_2^*) - cV - [(1-\varepsilon)\rho_1 k_1 T_1 + (1-f\varepsilon)\rho_2 k_2 T_2] V, \\
\dot{E} &= \lambda_E + \frac{b_E (T_1^* + T_2^*)}{T_1^* + T_2^* + K_b} E - \frac{d_E (T_1^* + T_2^*)}{T_1^* + T_2^* + K_d} E - \delta_E E,
\end{aligned}$$

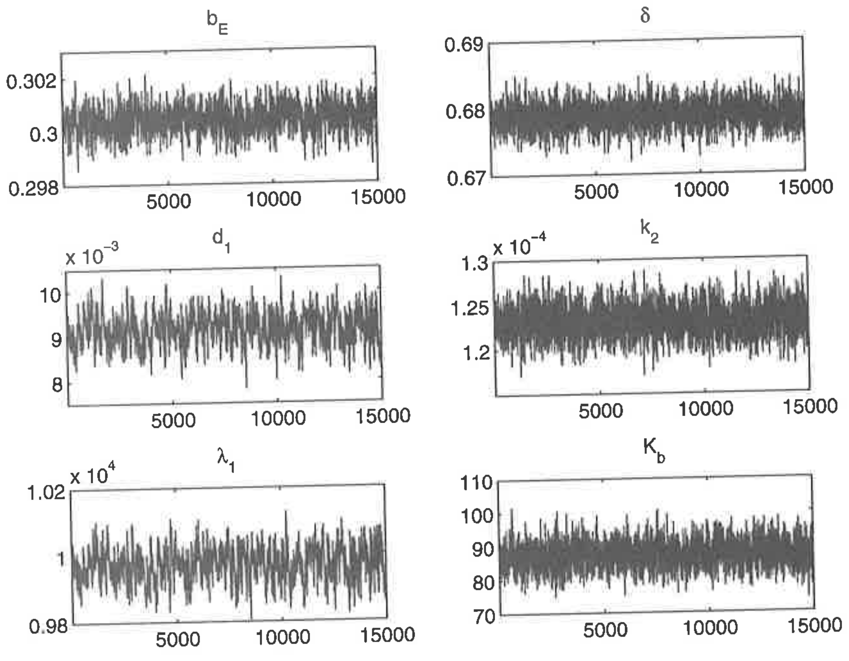


Figure 8.13. *Chains for* $Q = [b_E, \delta, d_1, k_2, \lambda_1, K_b]$.

developed in [2, 3] to provide a framework to investigate control strategies for HIV. As detailed in Example 3.3, T_1 and T_1^* represent the populations of uninfected and infected T-lymphocytes, T_2 and T_2^* are corresponding macrophage populations, and V, E denote the populations of free virus and immune effector cells.

To construct synthetic data for all six states, we add noise to model solutions computed using the parameter values reported in [3]. We note that clinical data comprised of the total number $T_1 + T_1^*$ of T-lymphocytes and viral load V can be found in [3].

For this example, we used the DRAM algorithm to construct chains and densities for the parameters $Q = [b_E, \delta, d_1, k_2, \lambda_1, K_b]$ with the remaining parameters fixed at the values reported in [3]. After a burn-in period of 5000 iterates, the parameter chains, densities, and joint sample points obtained using 15,000 DRAM iterates are plotted in Figures 8.13–8.15. We note that this is a relatively short burn-in period despite the fact that parameter values vary over eight orders of magnitude. It is observed from the pairwise joint sample plots in Figure 8.15 that k_2 and δ are clearly correlated, as are λ_1 and d_1. The fact that the parameters are not mutually independent proves important when we revisit this problem in Example 9.14, where we discuss propagation of uncertainty in models.

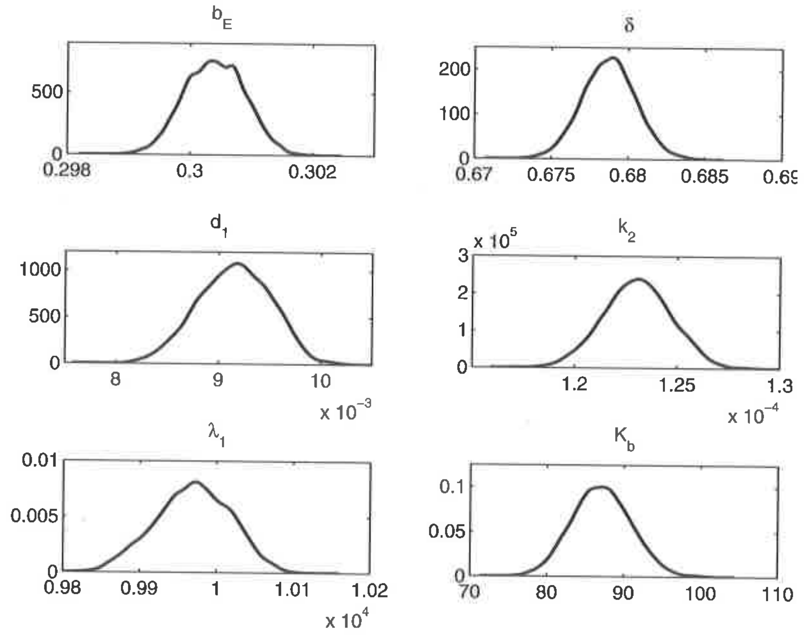


Figure 8.14. *Marginal densities for* $Q = [b_E, \delta, d_1, k_2, \lambda_1, K_b]$.

8.7 DiffeRential Evolution Adaptive Metropolis (DREAM)

It was illustrated in Section 8.3 that the scale and orientation of the proposal function critically affects the mixing and exploration of chains in standard random walk implementations of MCMC algorithms. The DRAM algorithm of Section 8.6 significantly improves performance through two mechanisms: adaptation updates the chain covariance matrix as information is obtained about the posterior density, and delayed rejection alters the proposal function in a predefined manner to improve mixing. For many models, that is sufficiently efficient for constructing parameter densities that can subsequently be employed for quantifying uncertainties in model responses or QoI.

However, there are various regimes for which DRAM algorithms are often not efficient. These include problems in which posterior densities are multimodal, are highly complex, or have heavy tails. For these cases, the single DRAM chain will be slow to traverse the posterior, which can significantly diminish its efficiency. Moreover, the computational overhead associated with complex models—such as the weather, climate, hydrology, and nuclear reactor models discussed in Chapter 1—often preclude the construction of burned-in single chains, whereas one can often compute shorter parallel chains using massively parallel architectures.

With the framework discussed in Section 8.6, these issues have motivated the development of parallel chain versions of the adaptive Metropolis algorithms. In the interchain adaptation approach detailed in [230], independent parallel chains,

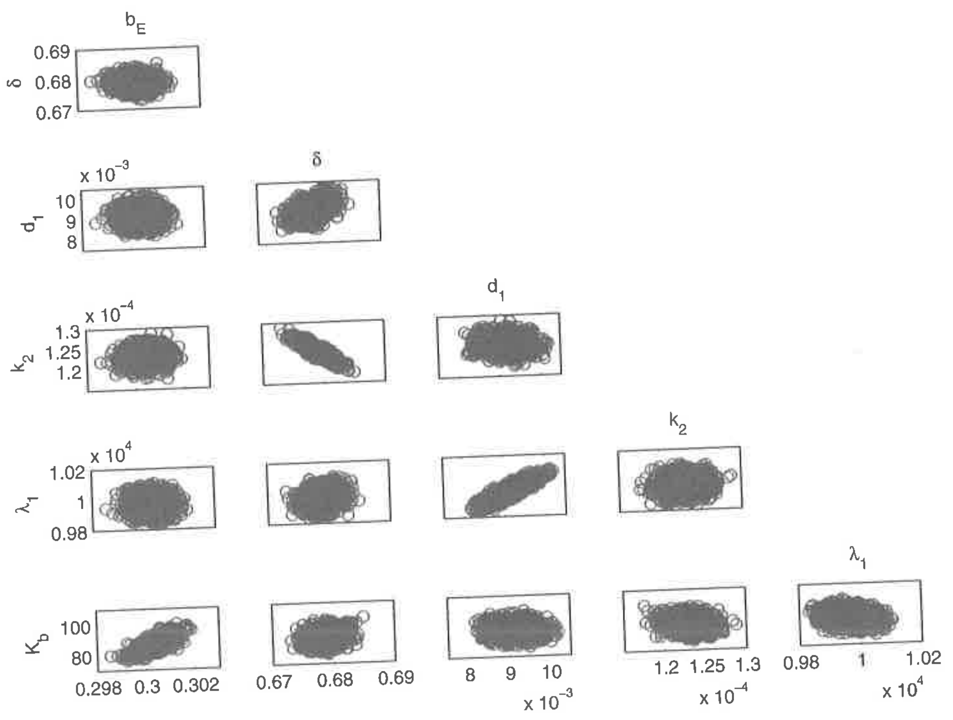


Figure 8.15. *Joint sample points for* $Q = [b_E, \delta, d_1, k_2, \lambda_1, K_b]$.

with early rejection mechanisms, are used to adapt the proposal function which in turn influences the mixing and exploration of future chain elements. This approach is highly parallelizable, improves the convergence of individual chains, and has been applied to climate models.

Differential evolution Markov chain (DE-MC) methods provide an alternative that can be more efficient for problems with multimodal or heavy tailed densities [246]. This approach can be summarized as follows. For p parameters, N chains q_i^k, $i = 1, \ldots, N$, are simultaneously run in parallel and the present population is stored in an $N \times p$ matrix X. Note that here the subscript designates the i^{th} chain rather than the i^{th} parameter component. In the original algorithm, candidates q_i^* were constructed by randomly choosing two chains from X, without replacement, and adding the weighted difference to q_i^k, that is,

$$q_i^* = q_i^k + \gamma \left(q_{i_1}^{k-1} - q_{i_2}^{k-1} \right) + e \quad , \; i_1 \neq i_2 \neq i,$$

for $i = 1, \ldots, N$. Typically, one takes e as realizations of $E \sim N(0, bI_p)$, where b is chosen smaller than the variance of the posterior. An optimal choice for the weight is $\gamma = 2.38/\sqrt{2p}$. During implementation, one often specifies $\gamma = 1$ at every 10th generation to permit direct jumping between modes. Candidates are then accepted

with probability $\alpha = \min(1, r)$, where r is the acceptance ratio given by (8.11) or (8.16).

The DE-MC algorithm differs from DRAM in the sense that it generates candidates based on current chain information stored in X rather than the covariance matrix V or chain covariance V_k defined in (8.19) which constitute the proposal function. The construction of q^* based on random members of the population facilitates the exploration of multimodal and heavy tailed posterior distributions and provides a mechanism for determining the appropriate scale, shape, and orientation of the proposal function. Further, the parallel chains in DE-MC algorithms learn from each other as compared with the DRAM implementation, where independent chains are used to adapt the proposal function. Theory establishing that Markov chains constructed in this manner have a unique stationary distribution and details regarding the implementation and performance of the DE-MC algorithm are provided in [246].

Further improvements in efficiency can be realized for many applications when similar evolution algorithms are combined with self-adaptive, randomized subspace sampling. This is the basis for the DiffeRential Evolution Adaptive Metropolis (DREAM) algorithm detailed in [261]. The candidates in this case are randomly generated using the algorithm

$$q_i^* = q_i^k + (I_p + f)\gamma(\delta, p') \left[\sum_{j=1}^{\delta} q_{i_1(j)}^{k-1} - \sum_{n=1}^{\delta} q_{i_2(n)}^{k-1} \right] + e,$$

where $i_1(j), i_2(n) \in \{1, \ldots, N\}$ satisfy $i_1(j) \neq i_2(n) \neq i$ for $j, n = 1, \ldots, N$. Here δ denotes the number of randomly sampled pairs and p' is the number of parameters that are jointly updated. Finally, f and e are realizations of uniform and normal random p-vectors; i.e., $F \sim \mathcal{U}_p(-b, b)$ and $E \sim N(0, b^* I_p)$, where b and b^* are small compared to the width of the posterior density.

For large parameter dimensions p, DREAM employs a random subspace sampling strategy that can decrease p' from its original values of p. This reduces the number of required parallel chains N so that, in theory, DREAM can run with $N < p$ as compared with $N = 2p$ required for DE-MC.

Convergence analysis and case studies for DREAM, DREAM$_{(D)}$, and MT-DREAM$_{(ZS)}$ are provided in [144, 259, 261]. Specifically, [261] illustrates applications where DREAM exhibits superior performance to DRAM and DE-MC. For moderate- to high-dimensional problems with computationally intensive codes, DREAM shares the advantage of the parallel DRAM algorithms since both can be implemented on massively parallel architectures. For example, the use of MT-DREAM$_{(ZS)}$ to perform Bayesian inference for 241 parameters in a hydrologic model is illustrated in [144].

Because DREAM is newer than DRAM, there are fewer available MATLAB toolboxes that are configured for general usage. However, that will certainly change in the next few years and readers are advised to incorporate both in their libraries of Bayesian parameter estimation routines for large, nonlinear engineering and scientific problems.

8.8 Notes and References

MCMC methods have proven highly successful for numerous applications quantified by data-based or statistical models. This is due in part to improved computational resources and the success of techniques such as the use of conjugate priors and Gibbs samplers, which permit parameter by parameter sampling when conditional posterior distributions can be reasonably approximated. However, the application of these techniques to engineering, science, and mathematical models was initially hampered by the following issues:

- the complexity and highly nonlinear dependence on parameters in models prohibited efficient use of conjugate priors and Gibbs samplers;
- appropriate statistical models and likelihood functions were difficult to formulate;
- static proposal functions were ineffective for exploring high-dimensional and complex posterior densities;
- the computational time required for codes associated with phenomena such as nonlinear, coupled, or high-dimensional PDEs prohibited burn-in.

These issues have been addressed in part by the development of algorithms such as DRAM and DREAM that have the adaptive capabilities to explore complex and multimodal posterior distributions and are amenable to implementation on massively parallel architectures. As a result, these algorithms are presently being employed for PDE models such as those employed for climate simulations. It is anticipated that their use will grow substantially as toolboxes evolve and the success of these and emerging algorithms is established.

We have focused primarily on Metropolis algorithms as a prelude for discussing DRAM and DREAM, and hence we have neglected several techniques, such as Gibbs samplers, which have proven highly successful in other contexts. The reader is referred to [92] for details regarding Gibbs samplers and the associated software package BUGS (Bayesian inference Using Gibbs Sampling) which can be implemented from within the statistical package R but is not yet available in MATLAB. Details about importance sampling can also be found in this reference. Sequential Monte Carlo (SMC) methods [75], which are also known as particle filters, can provide a more accurate alternative to extended or unscented Kalman filters for data assimilation when the number of samples is sufficiently large [79]. We refer the reader to [56, 240] for general overviews of Bayesian analysis and [15, 128, 244] for discussion regarding the use of Bayesian inference for parameter estimation. The interpretation of Tikhonov regularization in a Bayesian context is detailed in [27, 128].

8.9 Exercises

Exercise 8.1. By considering the cases $x < v, x = v$ and $x > v$, establish the relation

$$v \min(1, x/v) = \min(x, v) = x \min(1, v/x).$$

Exercise 8.2. Consider the steady state heat model of Example 8.12 and the temperature data v compiled in Table 3.2. Use DRAM to compute chains and marginal densities for the parameters $q = [\Phi, h]$. You should be able to reproduce the results in Example 8.12. Now compute the posterior density $\pi(q|v)$ directly using Bayes' relation (8.2). You can approximate the integral using tensored 1-D quadrature relations, as detailed in Chapter 11. Compare your posterior density with the joint density obtained using DRAM. Now numerically integrate $\pi(q|v)$ to construct marginal densities for Φ and h and compare to those constructed using DRAM.

Exercise 8.3. Repeat the computations of Example 8.12 using DRAM with the default proposal function $J(q^*|q^{k-1}) = N(q^{k-1}, D)$ rather than the choice $J(q^*|q^{k-1}) = N(q^{k-1}, V)$ used to obtain the reported results. Plot the first 200 iterates to show the initial burn-in period. Compare your final chain covariance matrix to that reported in the example.

Exercise 8.4. Verify the recursive relation (8.20).

Exercise 8.5. Show that V is singular if we try to estimate $q = [m, c, k]$ for the spring model

$$m\frac{d^2z}{dt^2} + c\frac{dz}{dt} + kz = 0,$$

$$z(0) = 2\ ,\ \frac{dz}{dt}(0) = -C.$$

Exercise 8.6. Here we are going to use DRAM to construct densities for parameters in the HIV model (8.24) detailed in Example 3.3 and illustrated in Example 8.13. Synthetic data is provided in the file `hiv-data` which can be downloaded from the website http://www.siam.org/books/cs12. The seven columns respectively contain the time and values for the six states $T_1, T_2, T_1^*, T_2^*, V$, and E measured every 5 days for 200 days. We are going to construct chains and densities for the parameters $q = [d_1, k_2, \delta, b_E]$.

You should start by writing a MATLAB code that uses `fminsearch` to optimize q based on this data. You can use the initial conditions (3.16) and remaining parameter values in Table 3.1.

Now use DRAM to compute densities for d_1, k_2, δ, and b_E. You should monitor your chains to ensure that they have burned-in or converged. Plot the chains, marginal densities, and pairwise scatterplots. We will revisit this problem in Exercise 9.7.