Bayesian Total Error Analysis For Hydrologic Models: Markov Chain Monte Carlo Methods To Evaluate The Posterior Distribution

George Kuczera, Dmitri Kavetski, Benjamin Renard and Mark Thyer

School of Engineering, University of Newcastle, New South Wales
Email: george.kuczera@newcastle.edu.au

Keywords: *Gibbs sampler, Bayesian hierarchical models, rainfall errors, rainfall-runoff models*

EXTENDED ABSTRACT

Calibration and prediction in conceptual rainfall-runoff (CRR) modelling is affected by the sampling and measurement uncertainty in the forcing/response data and by the structural error of the model conceptualisation. The Bayesian Total Error Analysis methodology (BATEA) offers a robust approach to deal with these multiple sources of uncertainty. The core idea is to pose the model calibration as a Bayesian hierarchical model with latent variables describing uncertainties in the data and the CRR model. This provides the opportunity to directly and comprehensively address all sources of uncertainty.

One objective of a BATEA analysis is to evaluate the posterior distribution of the model parameters and latent variables. This study reports on the application of Markov chain Monte Carlo (MCMC) approaches for sampling from the posterior distribution. The primary focus is the Gibbs sampler, traditionally the method of choice for hierarchical models because of its ease and robustness in handling hundreds, possibly thousands, of latent variables. However, in the case of CRR models with carryover of storage (in soil and groundwater stores) from one interval to the next, the Gibbs sampler rapidly becomes computationally intractable.

A full derivation of the Gibbs sampler is presented. It is shown that rigorous implementation of the Gibbs sampler in the presence of storage carryover involves a computational effort proportional to n^2 where n is the number of rainfall latent variables. Even for moderate n, the problem rapidly becomes intractable. A heuristic approximation that exploits the diminishing influence of initial conditions in the CRR model is presented to render the computational effort much closer to n. A case study shows that the heuristic Gibbs sampler produces a posterior distribution virtually indistinguishable from the posterior produced by the exact sampler. However, unlike the exact Gibbs sampler, the heuristic sampler is computationally efficient with effort proportional to the number of rainfall latent variables – see Figure 1. This result is of significance because it makes practical full analysis of CRR models with a large number of latent variables.

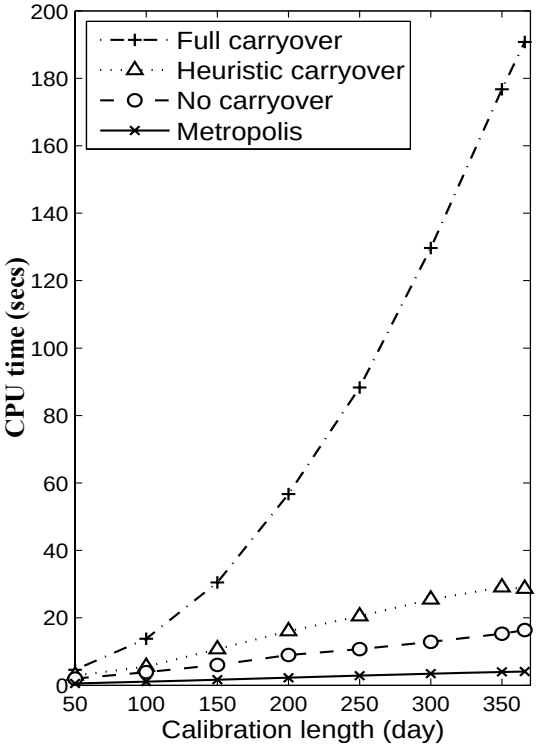

Figure 1. CPU time needed to simulate 1000 samples as a function of calibration record length.

1. INTRODUCTION

Catchment models simulate water balance dynamics at the catchment scale. Because of the significance of water in terrestrial ecosystems, catchment models are an integral part of virtually all environmental models formulated at the catchment scale and their applications range from catchment water and nutrient balances to biophysical models. This paper focuses on conceptual rainfall-runoff (CRR) models. An important, perhaps defining, feature of CRR models is that their parameters are not directly measurable and must be inferred ("calibrated") from the observed data. The advantage of this class of models is its ability to capture the dominant catchment dynamics while remaining parsimonious and computationally efficient.

Characterising the uncertainty in streamflow predicted by a CRR model has attracted the attention of hydrologists over many years. Yet in a recent review of CRR model calibration, Vrugt *et al*. (2005) note the lack of a robust framework that accounts for all sources of error (input, model and response error). This has a number of implications for CRR modelling: (i) quantifying the predictive uncertainty in streamflow and other model outputs is problematic; (ii) the regionalisation of CRR model parameters continues to be confounded by biases in the calibrated parameters and unreliable assessment of parameter uncertainty; and (iii) it is difficult to discriminate between competing CRR model hypotheses because poor model performance can "hide" behind the veil of ignorance about the sources of error.

Recently Kavetski *et al*. (2002, 2006a,b) and Kuczera *et al*. (2006) developed and illustrated a Bayesian total error analysis (BATEA) framework which discriminates between input, model and response errors. The key idea in the BATEA approach is to formulate the CRR model as a hierarchical model. This allows input and model errors to be treated as latent variables and enables exploitation of recent developments in Monte Carlo Markov chain (MCMC) methods.

The inclusion of latent variables introduces hundreds, possibly thousands, of variables requiring inference. Though counter intuitive, these latent variables do not overparameterize the model. Indeed hierarchical models involving large numbers of latent variables are routinely and efficiently solved using the Gibbs sampler (Chib and Greenberg, 1995) and public domain software BUGS (Gilks *et al.*, 1994). However, in the case of CRR models with carryover of storage (in soil and groundwater stores) from one interval to the next, the Gibbs sampler rapidly becomes computationally intractable.

The purpose of this paper is to carefully derive the Gibbs sampler for CRR models and then to develop approximations which restore its computational tractability without compromising the reliability of the inference.

2. AN OVERVIEW OF THE BATEA FRAMEWORK

Figure 2 presents a schematic of the BATEA hierarchical model. Without loss of generality, suppose a hydrologic time series is partitioned into n epochs $\{(t_i, t_{i+1}-1), i=1,..,n\}$ where t_i is the time step index corresponding to the beginning of the i^{th} epoch. The definition of an epoch is flexible. In the case of a CRR model, one reasonable epoch definition begins with a storm event and ends with a dry spell exceeding a minimum duration (Kuczera *et al*., 2006). The observed response time series for the i^{th} epoch is $\tilde{\mathbf{q}}_i = \{\tilde{\mathbf{q}}_t, t = t_i,..,t_{i+1}-1\}$, whereas $\mathbf{q}_i$ is the true response time series for the i^{th} epoch. The vectors $\tilde{\mathbf{x}}_i$ and $\mathbf{x}_i$ contain the observed and true forcing time series respectively for the i^{th} epoch.

The BATEA hypothesis of Kavetski *et al*. (2006a) assumes a function $g(\tilde{\mathbf{x}}_i, \boldsymbol{\varphi}_i)$ that maps the observed forcing $\tilde{\mathbf{x}}_i$ into the true forcing $\mathbf{x}_i$. The function g() accounts for the sampling and measurement error in the observed forcing. For example, Kavetski *et al*. (2006a) considered the special case of $\boldsymbol{\varphi}_i$ being a storm depth multiplier scalar, which yields the mapping $\mathbf{x}_i = \varphi_i \tilde{\mathbf{x}}_i$. The latent variable φ_i is assumed to vary from storm to storm and be a random realisation from the probability model with pdf $p(\boldsymbol{\varphi}|\boldsymbol{\alpha})$

$$\varphi_i \leftarrow p(\varphi|\boldsymbol{\alpha}) \qquad (1)$$

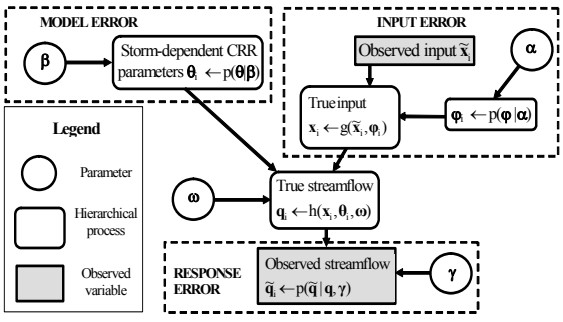

Figure 2. Schematic of the hierarchical BATEA model.

where $\boldsymbol{\alpha}$ is a vector of parameters describing the statistical properties of the input errors (e.g., the mean and variance of the multipliers).

Next it is assumed that for each epoch there exists a CRR model $h(\mathbf{x}_i, \boldsymbol{\theta}_i, \boldsymbol{\omega}, \mathbf{s}_i)$ that maps the true forcing $\mathbf{x}_i$ into the true response $\mathbf{q}$

$$\mathbf{q}_i \leftarrow h(\mathbf{x}_i, \boldsymbol{\theta}_i, \boldsymbol{\omega}, \mathbf{s}_i) \tag{2}$$

where $\boldsymbol{\omega}$ are the time-invariant CRR parameters, $\mathbf{s}$ is the vector of variables that represent storage (such as groundwater, soil moisture and stream storage) within the CRR model at the start of the epoch, and $\boldsymbol{\theta}_i$ is a set of epoch-specific CRR model parameters drawn from the hyper-distribution $p(\boldsymbol{\theta}|\boldsymbol{\beta})$

$$\boldsymbol{\theta}_i \leftarrow p(\boldsymbol{\theta}|\boldsymbol{\beta}) \tag{3}$$

where $\boldsymbol{\beta}$ are the CRR hyper-parameters (e.g., means and variances of the parameters). The epoch-dependent parameters are treated as latent (or hidden) variables in the same way as the input multipliers.

The storage vector is updated as follows

$$\mathbf{s}_{i+1} \leftarrow f(\mathbf{x}_i, \boldsymbol{\theta}_i, \boldsymbol{\omega}, \mathbf{s}_i) \tag{4}$$

This update is deterministic because conservation of mass applies to each storage element.

The observed response is corrupted by measurement error and is assumed to be distributed according to

$$\tilde{\mathbf{q}}_i \leftarrow p(\tilde{\mathbf{q}} \mid \mathbf{q}_i, \boldsymbol{\gamma}) \tag{5}$$

which is conditioned on the true discharge $\mathbf{q}$ and the parameter set $\boldsymbol{\gamma}$ that characterises the error process.

The hierarchical BATEA model is atypical of Bayesian hierarchical models, since the sampling of the true response $\mathbf{q}$ is not independent of earlier epochs. This complication arises because the time memory of CRR models induces a dependence between epochs: epoch-dependent parameters can affect model responses well beyond the current epoch. This dependence requires careful attention and precludes routine application of Bayesian hierarchical model packages.

2.1. BATEA Inference Problem

The primary objective of BATEA is to identify the parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\omega}$ and $\boldsymbol{\gamma}$ given the observed streamflow time series data $\tilde{\mathbf{Q}} = \{\tilde{\mathbf{q}}_i, i = 1,..,n\}$, the observed forcing time series $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_i, i = 1,..,n\}$ and any prior information. In the Bayesian framework this inference problem is described by the posterior pdf

$$\begin{aligned} &p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\gamma} \mid \tilde{\mathbf{Q}}, \tilde{\mathbf{X}}) = \\ &= \int p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\gamma}, \boldsymbol{\theta}_{1:n}, \boldsymbol{\varphi}_{1:n} \mid \tilde{\mathbf{Q}}, \tilde{\mathbf{X}}) \, d\boldsymbol{\theta}_{1:n} \, d\boldsymbol{\varphi}_{1:n} \end{aligned} \tag{6}$$

where $p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\gamma}, \boldsymbol{\theta}_{1:n}, \boldsymbol{\varphi}_{1:n} \mid \tilde{\mathbf{Q}}, \tilde{\mathbf{X}})$ is the full posterior pdf, $\boldsymbol{\theta}_{1:n} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n\}$ contains the sets of epoch-dependent CRR parameter realisations for all the storms, and $\boldsymbol{\varphi}_{1:n} = \{\boldsymbol{\varphi}_1, ..., \boldsymbol{\varphi}_n\}$. Direct evaluation of this integral is formidable due to its high dimensionality and strong nonlinearity.

Following Kavetski *et al.* (2002), it is advantageous (both statistically and computationally) to work directly with the full posterior probability distribution function (pdf). Inferring the latent variables enables one to test assumptions about hyperdistributions. This is vitally important because hypotheses about model structure and errors must be subject to empirical scrutiny. Moreover, the Gibbs sampler provides an elegant MCMC procedure for approximating the full posterior.

Before we consider the Gibbs sampler in detail the following changes are made to simplify the notation:

1. Since the latent variables of the input and structural error models are both associated with epochs, they are combined into $\boldsymbol{\theta}$, the set of epoch-dependent parameters or latent variables, with hyper-parameters $\boldsymbol{\beta}$.
2. The response measurement error parameters $\boldsymbol{\gamma}$ are assumed to be known. In the case of streamflow, this is a reasonable assumption for a well-maintained gauging station.

These changes yield the following posterior pdf:

$$\begin{aligned} &p(\boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\theta}_{1:n} \mid \tilde{\mathbf{Q}}, \tilde{\mathbf{X}}, \boldsymbol{\gamma}) \\ &= \frac{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\theta}_{1:n}, \tilde{\mathbf{X}}, \boldsymbol{\gamma}) p(\boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\theta}_{1:n}, \tilde{\mathbf{X}}, \boldsymbol{\gamma})}{p(\tilde{\mathbf{Q}}, \tilde{\mathbf{X}}, \boldsymbol{\gamma})} \\ &= \frac{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\omega}, \boldsymbol{\theta}_{1:n}, \tilde{\mathbf{X}}, \boldsymbol{\gamma}) p(\boldsymbol{\theta}_{1:n} \mid \boldsymbol{\beta}, \boldsymbol{\omega}, \tilde{\mathbf{X}}, \boldsymbol{\gamma}) p(\boldsymbol{\beta}, \boldsymbol{\omega} \mid \tilde{\mathbf{X}}, \boldsymbol{\gamma})}{p(\tilde{\mathbf{Q}}, \tilde{\mathbf{X}}, \boldsymbol{\gamma})} \\ &\propto p(\tilde{\mathbf{Q}} \mid \boldsymbol{\omega}, \boldsymbol{\theta}_{1:n}, \tilde{\mathbf{X}}, \boldsymbol{\gamma}) p(\boldsymbol{\theta}_{1:n} \mid \boldsymbol{\beta}) p(\boldsymbol{\beta}) p(\boldsymbol{\omega}) \end{aligned} \tag{7}$$

where $p(\tilde{\mathbf{Q}} \mid \boldsymbol{\omega}, \boldsymbol{\theta}_{1:n}, \tilde{\mathbf{X}}, \boldsymbol{\gamma})$ is the likelihood function (sampling distribution) of $\tilde{\mathbf{Q}}$ which, according to Figure 2, is independent from $\boldsymbol{\beta}$, $p(\boldsymbol{\theta}_{1:n} \mid \boldsymbol{\beta})$ is the hyper-distribution of $\boldsymbol{\theta}_{1:n}$ which only depends on $\boldsymbol{\beta}$, and $p(\boldsymbol{\omega})$ and $p(\boldsymbol{\beta})$ are prior pdfs.

3. GIBBS SAMPLER

The Metropolis-Hastings (MH) algorithm is a general MCMC procedure for sampling from multivariate distributions. Suppose $p(\boldsymbol{\theta})$ is the target distribution, the full posterior in this study, from which samples are required. The algorithm proceeds as follows (Chib and Greenberg, 1995):

Step 1: Initialize $\boldsymbol{\theta}$ with a suitable starting value $\boldsymbol{\theta}^0$. Set iteration counter t = 0.

Step 2: Sample a proposal vector $\boldsymbol{\theta}^*$ from a proposal or jump distribution with pdf $J(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$ where $\boldsymbol{\theta}^t$ is the value of the vector $\boldsymbol{\theta}$ at the t^{th} iteration.

Step 3: Evaluate the move ratio

$$r\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^t\right) = \frac{p\left(\boldsymbol{\theta}^*\right) J\left(\boldsymbol{\theta}^t \mid \boldsymbol{\theta}^*\right)}{p\left(\boldsymbol{\theta}^t\right) J\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^t\right)} \tag{8}$$

Step 4: Sample u from the uniform distribution U(0,1)

If $u < r(\boldsymbol{\theta}^* | \boldsymbol{\theta}^t)$ then $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^*$,

else $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t$

Step 5: Increment t. Check for convergence. If not converged go to step 1.

It is our experience with MH algorithms that the selection of the jump distribution, especially its covariance, is particularly important if the algorithm is to converge. In BATEA, the problem is made more challenging because the dimension of the jump distribution will grow with the number of latent variables.

We seek to avoid potential problems with high-dimensioned jump distributions by exploiting the hierarchical nature of the BATEA hypothesis in conjunction with block or alternating conditional MCMC sampling.

Chib and Greenberg (1995) provide a lucid exposition on block MCMC sampling which we will refer to as the Gibbs sampler. The hierarchical nature of the BATEA formulation shown in Figure 2 suggests the following three-block sampling approach. For the t^{th} iteration of the sampler the following three steps are performed:

Step 1: Sample $\boldsymbol{\beta}$ from its conditional posterior distribution $\boldsymbol{\beta}^{t+1} \leftarrow p(\boldsymbol{\beta} \mid \boldsymbol{\theta}_{1:n}^t, \boldsymbol{\omega}^t, \tilde{\mathbf{Q}}, \tilde{\mathbf{X}})$

The hierarchical dependence shown in Figure 2 enables the following simplification

$$p(\boldsymbol{\beta} \mid \boldsymbol{\theta}_{1:n}, \boldsymbol{\omega}^t, \tilde{\mathbf{Q}}, \tilde{\mathbf{X}}) = p(\boldsymbol{\beta} \mid \boldsymbol{\theta}_{1:n}) \tag{9}$$

For hyper pdfs of the Gaussian form, the posterior pdf $p(\boldsymbol{\beta}|\boldsymbol{\theta}_{1:n})$ is well known and can be directly sampled from (see Gelman *et al.*, 1995). Strictly speaking, when one can directly sample from the conditional posterior this is known as the Gibbs sampler.

Step 2: Sample $\boldsymbol{\theta}_{1:n}$ from its conditional posterior distribution $\boldsymbol{\theta}_{1:n}^{t+1} \leftarrow p(\boldsymbol{\theta}_{1:n} \mid \boldsymbol{\beta}^{t+1}, \boldsymbol{\omega}^t, \tilde{\mathbf{Q}}, \tilde{\mathbf{X}})$ where

$$p(\boldsymbol{\theta}_{1:n} \mid \boldsymbol{\beta}, \boldsymbol{\omega}, \tilde{\mathbf{Q}}, \tilde{\mathbf{X}}) = \frac{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\theta}_{1:n}, \boldsymbol{\omega}, \tilde{\mathbf{X}})\, p(\boldsymbol{\theta}_{1:n} \mid \boldsymbol{\beta})}{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\beta}, \boldsymbol{\omega}, \tilde{\mathbf{X}})} \tag{10}$$

Step 3: Sample $\boldsymbol{\omega}$ from its conditional posterior distribution $\boldsymbol{\omega}^t \leftarrow p(\boldsymbol{\omega} \mid \boldsymbol{\beta}^{t+1}, \boldsymbol{\theta}_{1:n}^{t+1}, \tilde{\mathbf{Q}}, \tilde{\mathbf{X}})$ where

$$p(\boldsymbol{\omega} \mid \tilde{\mathbf{Q}}, \boldsymbol{\theta}_{1:n}, \boldsymbol{\beta}, \tilde{\mathbf{X}}) = \frac{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\omega}, \boldsymbol{\theta}_{1:n}, \tilde{\mathbf{X}})\, p(\boldsymbol{\omega})}{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\theta}_{1:n}, \boldsymbol{\beta}, \tilde{\mathbf{X}})} \tag{11}$$

3.1. Epoch Block Metropolis-Hastings Sampler

It is highly unlikely that direct sampling of $\boldsymbol{\theta}_{1:n}$ from the conditional posterior in Step 2 is possible. In such cases a MH sampler will be required.

Since the latent variables $\boldsymbol{\theta}$ are sampled at the start of each epoch it is natural to employ a block sampling scheme based on epochs. For the j^{th} epoch and the t^{th} iteration of the Gibbs sampler the objective is to sample from the conditional posterior

$$\boldsymbol{\theta}_j^{t+1} \leftarrow p(\boldsymbol{\theta}_j \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\omega}^t, \tilde{\mathbf{Q}}, \tilde{\mathbf{X}}) \tag{12}$$

Exploiting the independence of the latent variables, the MH move ratio can be simplified to

$$
\begin{aligned}
&r\left(\boldsymbol{\theta}_j^* \middle| \boldsymbol{\theta}_j^t\right) \\
&= \frac{p(\boldsymbol{\theta}_j^* \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\omega}^t, \tilde{\mathbf{Q}}, \tilde{\mathbf{X}})\, J\left(\boldsymbol{\theta}_j^t \middle| \boldsymbol{\theta}_j^*\right)}{p(\boldsymbol{\theta}_j^t \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\omega}^t, \tilde{\mathbf{Q}}, \tilde{\mathbf{X}})\, J\left(\boldsymbol{\theta}_j^* \middle| \boldsymbol{\theta}_j^t\right)} \\
&= \frac{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_j^*, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\omega}^t, \tilde{\mathbf{X}})}{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_j^t, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\omega}^t, \tilde{\mathbf{X}})} \\
&\quad \times \frac{p(\boldsymbol{\theta}_j^* \mid \boldsymbol{\beta}^{t+1})}{p(\boldsymbol{\theta}_j^t \mid \boldsymbol{\beta}^{t+1})} \times \frac{J\left(\boldsymbol{\theta}_j^t \middle| \boldsymbol{\theta}_j^*\right)}{J\left(\boldsymbol{\theta}_j^* \middle| \boldsymbol{\theta}_j^t\right)}
\end{aligned}
\tag{13}
$$

3.2. Computation of Move Ratio

Evaluation of the move ratio (13) can be computationally very expensive. To see this it is best to expand the ratio of the likelihoods in the move ratio. For simplicity it is assumed that streamflow measurement error is Gaussian with mean $h(\mathbf{x}_k, \boldsymbol{\theta}_k, \boldsymbol{\omega}, \mathbf{s}_k)$ and variance σ_q^2. The pdf for the k[th] observed streamflow is expressed using the shorthand notation $N(\tilde{\mathbf{q}}_k \mid h(\mathbf{x}_k, \boldsymbol{\theta}_k, \boldsymbol{\omega}, \mathbf{s}_k), \sigma_q^2)$. The likelihood ratio thus becomes

$$
\begin{aligned}
&\frac{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_j^*, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\omega}^t, \tilde{\mathbf{X}})}{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_j^t, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\omega}^t, \tilde{\mathbf{X}})} \\
&= \frac{N(\tilde{\mathbf{q}}_j \mid h(\mathbf{x}_j, \boldsymbol{\theta}_j^*, \boldsymbol{\omega}^t, \mathbf{s}_j), \sigma_q^2)}{N(\tilde{\mathbf{q}}_j \mid h(\mathbf{x}_j, \boldsymbol{\theta}_j^t, \boldsymbol{\omega}^t, \mathbf{s}_j), \sigma_q^2)} \\
&\quad \times \prod_{k=j+1}^{n} \frac{N(\tilde{\mathbf{q}}_k \mid h(\mathbf{x}_k, \boldsymbol{\theta}_k^t, \boldsymbol{\omega}^t, \mathbf{s}_k), \sigma_q^2)}{N(\tilde{\mathbf{q}}_k \mid h(\mathbf{x}_k, \boldsymbol{\theta}_k^t, \boldsymbol{\omega}^t, \mathbf{s}_k), \sigma_q^2)}
\end{aligned}
\tag{14}
$$

When there is no storage carryover [that is, h() is independent of the storage], the ratio of the likelihoods simplifies to

$$
\begin{aligned}
&\frac{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_j^*, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\omega}^t, \tilde{\mathbf{X}})}{p(\tilde{\mathbf{Q}} \mid \boldsymbol{\theta}_{1:j-1}^{t+1}, \boldsymbol{\theta}_j^t, \boldsymbol{\theta}_{j+1:n}^t, \boldsymbol{\omega}^t, \tilde{\mathbf{X}})} \\
&= \frac{N(\tilde{\mathbf{q}}_j \mid h(\mathbf{x}_j, \boldsymbol{\theta}_j^*, \boldsymbol{\omega}^t, \mathbf{s}_j), \sigma_q^2)}{N(\tilde{\mathbf{q}}_j \mid h(\mathbf{x}_j, \boldsymbol{\theta}_j^t, \boldsymbol{\omega}^t, \mathbf{s}_j), \sigma_q^2)}
\end{aligned}
\tag{15}
$$

However, when there is carryover storage the second term in the likelihood ratio is no longer necessarily equal to one. This is because at the end of epoch j, the carryover storage to epoch j+1 will be different because the proposed latent variable $\boldsymbol{\theta}_j^*$ differs from the previously accepted latent variable $\boldsymbol{\theta}_j^t$; that is,

$$
\begin{aligned}
&\mathbf{s}_{j+1}^* = f(\mathbf{x}_j, \boldsymbol{\theta}_j^*, \boldsymbol{\omega}^t, \mathbf{s}_j) \\
&\neq \mathbf{s}_{j+1}^t = f(\mathbf{x}_j, \boldsymbol{\theta}_j^t, \boldsymbol{\omega}^t, \mathbf{s}_j)
\end{aligned}
\tag{16}
$$

The likelihood ratios given by equations (14) and (15) have very different computational demands. In the case of equation (15), two likelihood evaluations are necessary for each epoch yielding a total of 2n likelihood evaluations per iteration where n is the number of epochs. In the case of equation (14) which accounts for storage carryover, 2(n – j + 1) likelihood evaluations are required at the j[th] epoch. Over n epochs this amounts to n(n+1) likelihood evaluations! One would expect computational demands to become unmanageable even for calibrations involving only a few years of data and several hundred latent variables.

There are two ways to improve the computational efficiency of the Gibbs sampler:

1. Ignore the storage carryover. We will see this produces significant error in the posterior.
2. Monitor the ratio of the carryover likelihoods in equation (14) and use a heuristic rule to terminate the computation. The key idea is to exploit the fact that the effect of the different latent variables at epoch j will gradually diminish as the CRR model "forgets" the differences in the initial conditions expressed by equation (16). In this study a simple termination criterion is used, namely terminate when the log of the current ratio is less than some tolerance (0.001 in this study).

4. COMAPRISON OF SAMPLER PERFORMANCE

This section compares three different Gibbs samplers using the three-block structure described in Section 3: 1) Full carryover using equation (14); 2) No carryover using equation (15); and Heuristic carryover. Starting values for the Gibbs sampler were chosen as prior means. Within each block, the Gibbs sampler was implemented one variable at a time. Univariate Gaussian proposals were adaptively determined during the burn-in run.

The Gibbs sampler is also compared against a one-block MH sampler. In this case the Gaussian proposal is multivariate with high dimension owing to the presence of latent variables. The proposal covariance is computed using the results of the Gibbs sampler. Nonetheless, the one-block MH sampler represents the most efficient sampler in terms of computational effort and thus provides a benchmark by which to judge the Gibbs sampler performance.

To make the comparison meaningful a synthetic case study is performed so the MCMC sampler results can be compared against the true value of parameters. The case study is described fully in

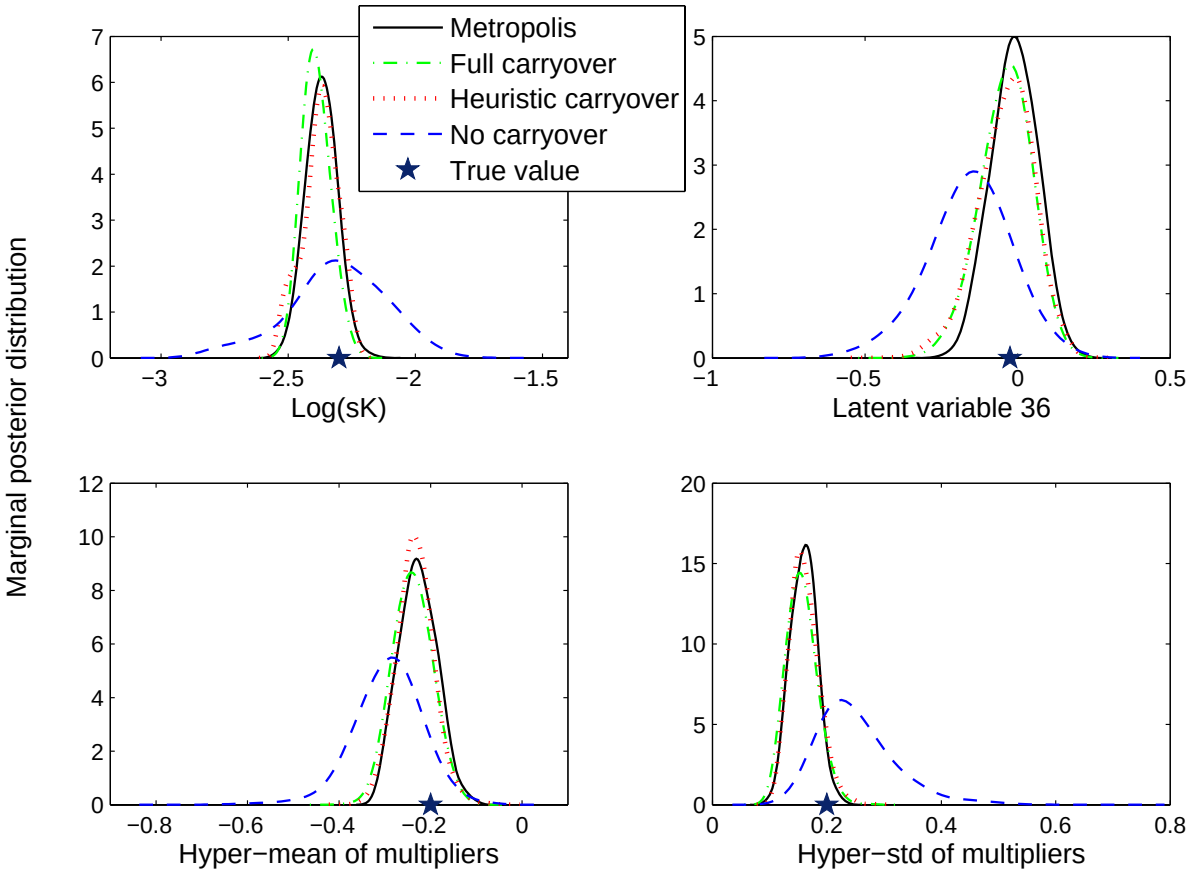

Figure 3. Posterior distributions for selected parameters derived by three different Gibbs samplers and a one-block Metropolis scheme.

Kuczera *et al*. (2006) and Renard *et al*. (2007). A time series of 366 daily rainfall and potential evapotranspiration for the Abercrombie catchment, NSW with area 2770 km^2 was used. These series define the 'true' input data. The logSPM CRR model (described in Kuczera *et al*., 2006) was used to simulate daily streamflow discharge using known parameters. The true streamflow is corrupted with independent normally distributed measurement error. Likewise the rainfall time series were corrupted by error multipliers drawn from a log-normal distribution

$$\log_e m \leftarrow N(-0.2, 0.2^2) \qquad (17)$$

Figure 3 presents the posterior distributions derived using the four MCMC samplers for the CRR parameter $\log_e sK$, one of the rainfall multipliers and mean and standard deviation of $\log_e m$. Figure 1 plots the computational time to produce 1000 samples as a function of the calibration length. Several conclusions can be drawn:

1. Of the three Gibbs samplers, the no-carryover sampler is computationally the most efficient. However, it produces a posterior distribution that is significantly in error. The distribution mis-specifies the mode and overestimates the posterior dispersion. It clearly is not a viable option when carryover storage occurs.

2. In contrast, the full and heuristic carryover Gibbs samplers and the one-block MH sampler yield virtually the same posterior distributions.

3. The computational performance of the full-carryover Gibbs sampler rapidly deteriorates with increasing number of latent variables. The increase in computing time is roughly linear for a given calibration length, but becomes quadratic as a function of calibration length. In contrast, the heuristic Gibbs sampler using the simple termination criterion only took two to three times longer than the no-carryover sampler.

4. The performance of the one-block MH sampler is exaggerated because the proposal covariance was set equal to the posterior covariance determined by the Gibbs sampler. However, what is of interest is the fact that the one-block MH sampler yielded virtually the same posterior distribution as the full carryover Gibbs sampler despite the high

dimension of the parameter space. This suggests that the posterior distribution must be moderately Gaussian. Indeed inspection of the posterior covariance suggests its structure may be amenable to simplification. If this proves to be the case, it may be preferable to use the very efficient one-block MH sample instead of the Gibbs sampler.

There is scope to further improve the efficiency of the heuristic sampler. For example, it is only necessary to evaluate the move ratio accurately when the move probability is less than 1. If the move ratio consistently stays above 1, then the proposal should be accepted even if the ratio has not stabilized. Likewise if the ratio remains consistently very small, then the proposal should be rejected with probability given by the current move ratio.

5. CONCLUSION

BATEA analysis for CRR models is based on a hierarchical Bayesian model which uses latent variables to describe the input and model errors affecting model predictions. A full Bayesian analysis enables inference of these latent variables and, importantly, permits direct scrutiny of the hypotheses describing input and model error. The Gibbs sampler has proven to be a robust tool for inferring the posterior distribution in hierarchical models. However, in the case of CRR models which are characterized by storages (such as soil and groundwater), a rigorous implementation of the Gibbs sampler becomes computationally intractable even for moderately small problems.

This study shows this problem can be overcome with a careful implementation of the Gibbs sampler. A heuristic approximation is used to derive a robust Gibbs sampler, meeting both requirements of computational efficiency and unbiasedness. The approximation exploits the fact that the effect of initial conditions in CRR models diminishes as the model moves forward in time. The heuristic sampler is a good compromise between a simulation scheme, which fully accounts for carryover between successive state storage values at the expense of intractable computing time, and a simulation scheme, which is fast because it ignores carryover but leads to mis-specification of the posterior distribution. It was found that the one-block Metropolis sampler was most efficient from a computational point of view, but requires a careful choice of the covariance of the Gaussian proposal. Given the Gibbs sampler was found to be efficient even with poorly specified starting values, this suggests the combined use of the Gibbs sampler with heuristic carryover and the one-block Metropolis sampler may yield the most efficient strategy to infer the BATEA posterior distribution involving hundreds, or even thousands, of latent variables.

6. ACKNOWLEDGEMENTS

Part of this work was funded by a grant from the Australia Research Council.

7. REFERENCES

Chib, S. and E. Greenberg (1995), Understanding the Metropolis-Hastings Algorithm, *The American Statistician*, 49(4), 327-335.

Gelman, A., Carlin, J.B., Stern, H.S. and Rubin, D.B. (1995), Bayesian data analysis, Chapman and Hall.

Gilks, W. R., A. Thomas, and D. J. Spiegelhalter (1994), A Language and Program for Complex Bayesian Modeling. *Statistician* 43:169-177.

Kavetski, D., S.W. Franks, and G. Kuczera, (2002), Confronting Input Uncertainty in Environmental Modelling, in "Calibration of Watershed Models", by Q. Duan, H. Gupta, S. Sorooshian, A. Rousseau, and R. Tourcotte (eds), Water Science and Application Series 6, American Geophysical Union, Washington DC, 49-68.

Kavetski, D., G. Kuczera, and S.W. Franks (2006a), Bayesian analysis of input uncertainty in hydrological modelling. I. Theory, *Water Resources Research*, vol. 42, W03407, doi:10.1029/2005WR004368.

Kavetski, D., G. Kuczera and S.W. Franks (2006b), Bayesian analysis of input uncertainty in hydrological modelling. II. Application, *Water Resources Research*, vol. 42, W03408, doi:10.1029/2005WR004376.

Kuczera, G., D. Kavetski, S.W. Franks and M. Thyer (2006), Towards a Bayesian total error analysis of conceptual rainfall-runoff models: Characterising model error using storm-dependent parameters, *Journal of Hydrology*, 331:1-2,161-177.

Renard, B., M. Thyer, G. Kuczera and D. Kavetski (2007), Bayesian total error analysis for hydrologic models: Sensitivity to error models, in *MODSIM Congress*, Christchurch, New Zealand.

Vrugt, J.A, C.G.H. Diks, H.V. Gupta, W. Bouten and J.M. Verstraten (2005), Improved treatment of uncertainty in hydrological modelling: Combining the strengths of global optimization and data assimilation, *Water Resources Research*, 41, W01017, doi:10.1029/2004WR0030059.