Corruption of accuracy and efficiency of Markov chain Monte Carlo simulation by inaccurate numerical implementation of conceptual hydrologic models

G. Schoups,[1] J. A. Vrugt,[2,3,4] F. Fenicia,[1,5] and N. C. van de Giesen[1]

Received 17 September 2009; revised 27 May 2010; accepted 10 June 2010; published 20 October 2010.

[1] Conceptual rainfall-runoff models have traditionally been applied without paying much attention to numerical errors induced by temporal integration of water balance dynamics. Reliance on first-order, explicit, fixed-step integration methods leads to computationally cheap simulation models that are easy to implement. Computational speed is especially desirable for estimating parameter and predictive uncertainty using Markov chain Monte Carlo (MCMC) methods. Confirming earlier work of Kavetski et al. (2003), we show here that the computational speed of first-order, explicit, fixed-step integration methods comes at a cost: for a case study with a spatially lumped conceptual rainfall-runoff model, it introduces artificial bimodality in the marginal posterior parameter distributions, which is not present in numerically accurate implementations of the same model. The resulting effects on MCMC simulation include (1) inconsistent estimates of posterior parameter and predictive distributions, (2) poor performance and slow convergence of the MCMC algorithm, and (3) unreliable convergence diagnosis using the Gelman-Rubin statistic. We studied several alternative numerical implementations to remedy these problems, including various adaptive-step finite difference schemes and an operator splitting method. Our results show that adaptive-step, second-order methods, based on either explicit finite differencing or operator splitting with analytical integration, provide the best alternative for accurate and efficient MCMC simulation. Fixed-step or adaptive-step implicit methods may also be used for increased accuracy, but they cannot match the efficiency of adaptive-step explicit finite differencing or operator splitting. Of the latter two, explicit finite differencing is more generally applicable and is preferred if the individual hydrologic flux laws cannot be integrated analytically, as the splitting method then loses its advantage.

Citation: Schoups, G., J. A. Vrugt, F. Fenicia, and N. C. van de Giesen (2010), Corruption of accuracy and efficiency of Markov chain Monte Carlo simulation by inaccurate numerical implementation of conceptual hydrologic models, *Water Resour. Res.*, *46*, W10530, doi:10.1029/2009WR008648.

1. Introduction

[2] Accurate assessment of the parameters and predictive uncertainty of hydrologic models is an important aspect of any hydrologic modeling application. It provides insights into the adequateness of the model, and indicates whether the data contain enough information to identify the model parameters [*Vrugt et al.*, 2006]. For example, strong parameter correlations may point to the need for additional data collection aimed at reducing parameter uncertainty [*Kuczera*, 1983]. Accurate parameter uncertainty estimation is also crucial for developing regionalization relationships between observable basin characteristics and unobservable model parameters, with the aim of extrapolation of hydrologic parameters to ungauged basins [*Wagener et al.*, 2007].

[3] A powerful and flexible method for estimating the parameter uncertainty of dynamic models is based on Markov chain Monte Carlo (MCMC) sampling. These algorithms are especially useful for the parameter inference of nonlinear hydrologic models, for which analytical expressions of the posterior parameter distributions are not available [*Kuczera and Parent*, 1998; *Bates and Campbell*, 2001; *Vrugt et al.*, 2003; *Engeland et al.*, 2005; *Smith and Marshall*, 2008]. Much research has focused on improving the efficiency and convergence of MCMC samplers to efficiently sample from high-dimensional parameter distributions [*ter Braak and Vrugt*, 2008; *Vrugt et al.*, 2009; *Kuczera et al.*, 2010]. However, much less attention has been paid to the effects on the MCMC algorithmic performance of inaccurate numerical implementation of the hydrologic models used in the analysis.

[4] Many existing conceptual hydrologic models use relatively inaccurate explicit time integration of the water

[1] Department of Water Management, Delft University of Technology, Delft, Netherlands.

[2] Department of Civil and Environmental Engineering, University of California, Irvine, California, USA.

[3] Also at Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico, USA.

[4] Also at Institute for Biodiversity and Ecosystem Dynamics, University of Amsterdam, Amsterdam, Netherlands.

[5] Also at Centre de Recherche Public Gabriel Lippmann, Belvaux, Luxembourg.

0043-1397/10/2009WR008648

balance dynamics, which results in simple and fast models [*Singh and Woolhiser*, 2002]. An underlying assumption has been that numerical errors are small relative to other model errors. However, work by Kavetski and coworkers [*Kavetski et al.*, 2003; *Kavetski et al.*, 2006a; *Kavetski and Kuczera*, 2007; *Clark and Kavetski*, 2010; *Kavetski and Clark*, 2010] has drawn attention to the importance of accurate numerical implementation of conceptual hydrologic models. In particular, these studies have highlighted problems with numerical implementation of several popular conceptual rainfall-runoff models, caused by explicit fixed-step temporal integration of the water balance equations [*Kavetski et al.*, 2003] and threshold-type flux-storage relationships [*Kavetski and Kuczera*, 2007]. The resulting numerical errors may be significant and are responsible for irregular parameter response surfaces with both microscale (e.g., nonsmoothness) and macroscale (e.g., secondary optima) features that often are removed when the same model is implemented with a numerically accurate method. Such deformations of the parameter response surface necessarily affect parameter and predictive inference and sensitivity analysis of the hydrologic model [*Kavetski and Clark*, 2010]. The need for formal and accurate numerical implementation of models has also been recognized and adopted in the wider environmental modeling literature [e.g., *Parkhurst and Appelo*, 1999; *Kavetski et al.*, 2001; *Cox and Whitehead*, 2005; *Qu and Duffy*, 2007].

[5] To the extent that inaccurate numerical implementation of hydrologic models induces changes in the parameter response surface, it may lead to errors in estimated optimal parameter values and estimated parameter uncertainty. Whereas gradient-based methods clearly are affected by poor numerical implementation of hydrologic models, e.g., causing premature convergence to a secondary optimum or more generally creating problems with computing gradients on nonsmooth surfaces [e.g., *Kavetski et al.*, 2006a], the effects on parameter and predictive inference using MCMC-based methods is less obvious and has not been studied in detail. In a recent study, *Kavetski and Clark*, [2010] extensively compared several numerical integration methods for a range of hydrologic basins using various hydrologic model structures. Although their focus was on improved numerical methods for gradient-based optimization and sensitivity analysis, they also showed that marginal posterior parameter distributions obtained by MCMC simulation may be significantly affected by inaccurate numerical implementation based on fixed-step explicit, and even implicit, methods.

[6] In this paper, we expand on their work and specifically focus on the effect of inaccurate numerical integration methods, typical of conceptual rainfall-runoff models, on the accuracy and efficiency of MCMC-based estimation of parameter and predictive uncertainty. We consider a single rainfall-runoff case study and apply a state-of-the-art MCMC algorithm [*Vrugt et al.*, 2009] for parameter and predictive inference of a simple spatially lumped hydrologic model. This setup can be considered an optimistic scenario, in that any effects on accuracy and efficiency observed in this case will likely be amplified when less powerful MCMC algorithms are used in combination with more complex models. The paper is organized as follows. Section 2 briefly describes the spatially lumped hydrologic model used in this study. Sections 3 and 4 summarize the numerical integration methods and the MCMC algorithm used, respectively. Results are presented in section 5, followed by a discussion of the implications of our work (section 6) and conclusions (section 7).

2. Hydrologic Model

[7] We use a spatially lumped hydrologic model to simulate daily rainfall-runoff processes. Our modeling approach is derived from the FLEX model of *Fenicia et al.* [2007]. The model consists of an unsaturated zone water balance equation, which is used to partition rainfall into evaporation, runoff, and percolation. Runoff and percolation are then each routed through a linear reservoir. The water balance for the unsaturated zone is

$$S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = P_e - Q_f - Q_e - Q_s, \tag{1}$$

where S_r is relative storage (= $S/S_{\max}$), S is total storage (L), $S_{\max}$ is maximum storage capacity (L), t is time (T), P_e is effective rainfall rate (L/T), Q_f is runoff (L/T), Q_e is actual evaporation rate (L/T), and Q_s is percolation rate (L/T). Effective rainfall consists of that part of the rainfall that is not intercepted by vegetation. For the results in this paper, we assume that interception is negligible, such that P_e is equal to the recorded rainfall rate. The three remaining fluxes in (1) are parameterized as single-valued, monotonic functions of relative storage:

$$\begin{aligned} Q_f &= P_e f(S_r;\alpha_F),\\ Q_e &= E_p f(S_r;\alpha_E),\\ Q_s &= Q_{s\max} f(S_r;\alpha_S), \end{aligned} \tag{2}$$

where E_p is potential evaporation rate (L/T), $Q_{s\max}$ is maximum percolation rate (L/T), and α_F, α_E, and α_S are process-specific parameters. The flux function f is assumed to take the form

$$f(S_r;\alpha) = \frac{1-e^{-\alpha S_r}}{1-e^{-\alpha}}. \tag{3}$$

This function is monotonically increasing from 0 to 1, as S_r increases from 0 to 1. Large positive (negative) values for α result in a fast (slow) increase to 1, whereas $\alpha \rightarrow 0$ gives a linear increase. The amount of water that percolates downward, Q_s, is routed through a linear reservoir, characterized by a time constant K_S (T), whereas runoff Q_f is routed through a linear reservoir with time constant K_F (T).

3. Numerical Integration Methods

[8] As the routing reservoirs are linear, their water balances are readily computed by analytical integration. However, the unsaturated zone dynamics are nonlinear and need to be computed numerically. We consider two general approaches: one based on finite difference approximations (section 3.1) and the other using operator splitting techniques (section 3.2). Section 3.3 presents adaptive time stepping methods used in this paper.

3.1. Finite Differencing

[9] The standard approach to numerical integration of equation (1) is based on finite differencing [*Butcher*, 2008]. The simplest approach uses the explicit Euler method to

Table 1. Overview of Numerical Integration Methods Used

Symbol	Description	Accuracy	Time Step
S1	Sequential splitting with explicit Euler	First order	Fixed (daily)
S2	Symmetrically weighted sequential splitting with analytical integration	Second order	Adaptive
E1	Explicit Euler	First order	Fixed (daily)
E2	Explicit trapezoidal rule	Second order	Adaptive
I1	Implicit Euler	First order	Fixed (daily)
I2	Implicit trapezoidal rule	Second order	Adaptive
REF	Explicit Runge-Kutta	Fifth order	Adaptive

approximate the derivative on the left-hand side. This leads to fast and easily implementable code, which explains its popularity in conceptual environmental models [*Singh and Woolhiser*, 2002]. Disadvantages of the explicit Euler method are its low accuracy and its conditional stability, which may cause significant numerical errors, especially if implemented with fixed time stepping. Alternatives are the use of implicit integration methods, e.g., the implicit Euler method used by *Kavetski et al.* [2006a], or more accurate explicit methods, e.g., the second-order accurate explicit method with adaptive time stepping used by *Kavetski et al.* [2003] to integrate the rainfall-runoff model TOPMODEL.

[10] In this paper we consider the following finite difference approximation of equation (1):

$$S_{r,t} = S_{r,t_0} + \Delta t\left[\theta g\left(S^{*}_{r,t}\right) + (1-\theta) g\left(S_{r,t_0}\right)\right], \qquad (4)$$

where $S_{r,t}$ is relative storage at time t, with $t = t_0 + \Delta t$ and time step Δt. Function g follows from equations (1)–(3) and is given by $g(S_r) = (P_e - Q_f - Q_e - Q_e)/S_{\max}$. Depending on the choice for θ and $S^{*}_{r,t}$, equation (4) reduces to several well-known numerical schemes (Table 1): (1) for $\theta = 0$ we obtain the first-order accurate, explicit Euler method (denoted as E1), (2) setting $\theta = 1$ and $S^{*}_{r,t} = S_{r,t}$ results in the first-order accurate, implicit Euler method (I1), (3) $\theta = 0.5$ and $S^{*}_{r,t} = S_{r,t_0} + \Delta t g(t_0, S_{r,t_0})$ yields the second-order accurate, explicit trapezoidal rule (E2), and (4) setting $\theta = 0.5$ and $S^{*}_{r,t} = S_{r,t}$ results in the second-order accurate implicit trapezoidal rule (I2). These methods are all standard and are discussed in detail in any reference book on ordinary differential equations [e.g., *Butcher*, 2008]. The two implicit methods (I1 and I2) require solution of a nonlinear problem at each time step, which is done here using Newton-Raphson linearization with bracketing and bisection [*Press et al.*, 1990]. Tolerance for nonlinear iterations is set to 10^{-8}.

3.2. Operator Splitting

[11] An alternative approach to integrating ordinary differential equations, such as the water balance equation in (1), is based on operator splitting [*McLachlan and Quispel*, 2002]. The basic idea is to sequentially split the right-hand-side fluxes in equation (1) into two or more parts that are easier to integrate than the original problem (“divide and conquer”). The solution of each subproblem then serves as the initial condition for the next subproblem. The method has a long history for solving flow and reactive transport problems, by separately considering advective, dispersive, and reactive terms [e.g., *Steefel and MacQuarrie*, 1996].

[12] The splitting technique has also been applied in conceptual rainfall-runoff models, although it is often not recognized as such. A typical example is the sequential computation of runoff, evaporation, and percolation processes, as implemented in one form or another in, e.g., the VIC model [*Wood et al.*, 1992], the GR4J model [*Perrin et al.*, 2003], and the HYMOD model [*Vrugt et al.*, 2008]. Indeed, an obvious choice for splitting equation (1) is to consider storage changes due to different hydrologic processes (infiltration/runoff, evaporation, and percolation) as three separate problems:

$$S_{\max}\frac{dS_r}{dt} = P_e - Q_f, \quad S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = -Q_e, \quad S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = -Q_s. \qquad (5)$$

In order to find the approximate change in water storage over time step Δt due to these three processes, we specify (1) the computation sequence of these processes and (2) methods for integrating each subproblem. Note that splitting the original ordinary differential equation (ODE) introduces numerical errors, i.e., splitting errors, in addition to any truncation errors resulting from numerical solution of the various subproblems [*Csomos and Farago*, 2008].

[13] Several splitting techniques may be used, as reviewed by *McLachlan and Quispel* [2002] and *Csomos et al.* [2005]. The most straightforward, but also least accurate, approach uses sequential splitting. Applied to our case, equation (5), it means that we compute overall change in storage during a time step Δt as a sequence of three consecutive steps. One possible sequence is

$$\begin{aligned}
&S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = P_e - Q_f, \quad S_{r,t_0}\xrightarrow{\Delta t}S_{r,1},\\
&S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = -Q_e, \quad S_{r,1}\xrightarrow{\Delta t}S_{r,2},\\
&S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = -Q_s, \quad S_{r,2}\xrightarrow{\Delta t}S_{r,t},
\end{aligned} \qquad (6)$$

where the solution of each subproblem serves as the initial condition for the next one. The splitting error caused by sequential splitting is $O(\Delta t^2)$, and therefore this scheme is first-order accurate [*Csomos et al.*, 2005]. Second-order accuracy can be achieved by considering more elaborate splitting schemes. A popular example is the method proposed by *Strang* [1968], which, applied to equations (1) and (6), advances S_{r,t_0} to $S_{r,t}$ in five consecutive steps:

$$\begin{aligned}
&S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = P_e - Q_f, \quad S_{r,t_0}\xrightarrow{\Delta t/2}S_{r,1},\\
&S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = -Q_e, \quad S_{r,1}\xrightarrow{\Delta t/2}S_{r,2},\\
&S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = -Q_s, \quad S_{r,2}\xrightarrow{\Delta t}S_{r,3},\\
&S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = -Q_e, \quad S_{r,3}\xrightarrow{\Delta t/2}S_{r,4},\\
&S_{\max}\frac{\mathrm{d}S_r}{\mathrm{d}t} = P_e - Q_f, \quad S_{r,4}\xrightarrow{\Delta t/2}S_{r,t}.
\end{aligned} \qquad (7)$$

[14] Another second-order splitting scheme is called symmetrically weighted sequential (SWS) splitting, as discussed by *Csomos et al.* [2005]. Applied to equation (1), it solves the sequence in (6) twice, once using the order specified in (6), i.e., infiltration/runoff → evaporation → percolation, and

Table 2. Expressions for Coefficients A and B in the General Ordinary Differential Equation $\mathrm{d}S_r/\mathrm{d}t = A + Be^{-\alpha S_r}$ With Analytical Solution $S_r = (1/\alpha)\log[(e^{\alpha S_{r,0}} + B/A)e^{\alpha A \Delta t} - (B/A)]$ When Integrated From $S_{r,0}$ to S_r Over Time Step Δt

Process	A	B
Infiltration/runoff	$-\dfrac{P_e e^{-\alpha_F}}{S_{\max}(1-e^{-\alpha_F})}$	$\dfrac{P_e}{S_{\max}(1-e^{-\alpha_F})}$
Evaporation	$\dfrac{-E_p}{S_{\max}(1-e^{-\alpha_E})}$	$\dfrac{E_p}{S_{\max}(1-e^{-\alpha_E})}$
Percolation	$\dfrac{-Q_{s\max}}{S_{\max}(1-e^{-\alpha_S})}$	$\dfrac{Q_{s\max}}{S_{\max}(1-e^{-\alpha_S})}$

again using the reverse order, i.e., percolation ≥ evaporation ≥ infiltration/runoff. The overall solution after one time step is then obtained as the average of the solutions from these two sequences. Therefore, SWS splitting applied to our problem requires six steps in each time step (three for each sequence), whereas Strang splitting as in equation (7) only requires five steps. However, as will be discussed in the next section, SWS splitting has an important advantage when implemented with adaptive time stepping, and is adopted here instead of Strang splitting.

[15] Following our choice of finite difference methods in section 3.1, we compare two splitting methods. The first one uses sequential splitting as in (6), combined with the explicit Euler method to compute each subproblem. This approach is first-order accurate and will be denoted by S1 (Table 1). The second approach uses SWS splitting, combined with analytical integration of each subproblem in (6), resulting in a second-order accurate scheme (S2). With the exponential flux law in (3), the individual ODEs in (6) can be written in the general form $\mathrm{d}S_r/\mathrm{d}t = A + Be^{-\alpha S_r}$, yielding analytical solutions for S_r shown in Table 2. These analytical expressions provide, in principle, error-free solutions for the individual ODEs in (6). However, as pointed out by *Kavetski and Kuczera* [2007], implementation of such expressions may introduce cancellation errors due to poor scaling of the various exponential terms. The appendix details how the analytical solutions in Table 2 were implemented to minimize such problems.

[16] Note that the ordered sequences proposed in equations (6) and (7) are just one possible strategy for splitting; i.e., the three individual processes could be solved in a different order. It is not our goal here to test all possible combinations. However, for the case study discussed in this paper, preliminary tests with different sequences (in particular, replacing the sequence infiltration/runoff → evaporation → percolation with either infiltration/runoff → percolation → evaporation or percolation → evaporation → infiltration/runoff) suggested negligible effect on the conclusions reached in this paper.

3.3. Adaptive Time Stepping

[17] Time steps Δt in the numerical schemes discussed above (operator splitting and finite differencing) can be fixed, e.g., by setting them equal to a fraction of the time scale of the boundary forcing, or they can be changed automatically. The advantage of adaptive time stepping is that a predefined numerical accuracy can be achieved independent of the changing dynamics of the simulated processes [*Butcher*, 2008]. Achieving this predefined accuracy translates into taking small steps when flow is highly dynamic during and after storm events, and allowing bigger steps when flow and storage changes occur much more gradually, e.g., during streamflow recession. Adaptive time stepping requires two ingredients: first, a method to estimate the local numerical errors incurred during an individual step Δt, and, second, a method to adjust the time step.

[18] A standard method for evaluating local numerical errors is by comparing the solution to a higher-order, more accurate solution [*Butcher*, 2008]. This is also the approach adopted here: as shown in Table 1, each of the first-order methods (E1, I1, S1) has a second-order equivalent (E2, I2, S2), which we use to estimate the local numerical error ε of the corresponding first-order method, $\varepsilon = |S_r^I - S_r^{II}|$, where I and II stand for first and second order, respectively. Hence, for the second-order finite difference methods (E2, I2), error ε corresponds to the local (Taylor series) truncation error of the corresponding first-order method (E1, I1):

$$\varepsilon = 0.5\Delta t \left| g\left(S_{r,t_0}\right) - g\left(S^*_{r,t}\right) \right|, \tag{8}$$

with $S^*_{r,t}$ as defined in (4). For the second-order SWS splitting method with analytical integration (S2), we obtain ε by comparing the solution not to S1 but to first-order sequential splitting with analytical integration. In this case, error ε estimates the local splitting error of the first-order method, as there are no truncation errors due to analytical integration:

$$\varepsilon = \left| S^{(f)}_{r,t} - 0.5\left(S^{(f)}_{r,t} + S^{(b)}_{r,t}\right) \right|, \tag{9}$$

with $S^{(f)}_{r,t}$ and $S^{(b)}_{r,t}$ referring to forward and backward sequences in (6). Note here the advantage of SWS over Strang splitting, as it requires no extra computational cost for estimating splitting error ε.

[19] The current time step is accepted when error ε is less than a specified tolerance value τ. Otherwise, computation of $S_{r,t}$ is discarded and the simulation proceeds from S_{r,t_0} with a smaller time step. In either case, a new time step is computed using the following method [*Butcher*, 2008]:

$$\Delta t_{\text{new}} = 0.9\Delta t_{\text{old}} \left(\frac{\tau}{\varepsilon}\right)^{1/(p+1)}, \tag{10}$$

where Δt_{old} is the previous time step, Δt_{new} is the new time step, p is the order of the numerical scheme for which error ε is estimated (i.e., p = 1), and 0.9 acts as a safety factor. Following *Butcher* [2008], we limit consecutive increases and decreases in the time step by a factor of 2 or less. As indicated in Table 1, in this paper adaptive time stepping is used for the second-order methods, whereas for the first-order methods a fixed daily time step is used, corresponding to the daily time scale of the forcing data in the case study (see section 5).

[20] We note that these adaptive ODE integration methods are quite standard in the (numerical) literature and have also been implemented in a range of environmental models, including geochemical and biochemical models describing kinetic reactions [*Parkhurst and Appelo*, 1999], water quality modeling [*Cox and Whitehead*, 2005], river basin modeling

Table 3. Hydrologic Model Parameters With Prior Minimum and Maximum Values[a]

Parameter	Description	Minimum	Maximum
$S_{\max}$	Maximum storage capacity (mm)	0	5000
$Q_{s\max}$	Maximum percolation rate (mm/d)	0	100
α_E	Exponential parameter for evaporation	0	100
α_F	Exponential parameter for fast runoff	−100	100
K_F	Response time of fast runoff (days)	0	10
K_S	Response time of slow runoff (days)	0	500
σ	Residual standard deviation (mm)	0	1

[a]Parameter α_S is kept constant at a value of 1×10^{-6} (i.e., percolation rate varies linearly with storage).

[*Qu and Duffy*, 2007], and variably saturated groundwater flow models [*Kavetski et al.*, 2001].

4. MCMC Simulation

[21] Markov chain Monte Carlo (MCMC) simulation is used to estimate the uncertainty of the hydrologic model parameters θ after observing data x. This is done by random sampling from the posterior parameter density $\pi(\theta)$ given by the Bayes rule,

$$\pi(\theta) \propto f(x|\theta)\pi_0(\theta), \tag{11}$$

where $\pi_0(\theta)$ is the prior parameter density, reflecting uncertainty about θ before data x was available, and $f(x|\theta)$ is a probability density function (pdf) describing the likelihood of the data x being generated by parameter set θ, with the means of this pdf typically simulated by a deterministic hydrologic model (as in section 2). Many different MCMC algorithms have been developed in the past decade to efficiently generate samples from the posterior distribution, $\pi(\theta)$. These algorithms differ in their search and adaptation strategy, but always implement the same two basic steps [*Robert and Casella*, 2004]. First, a candidate point is generated in each individual Markov chain by sampling from a specified proposal density, $q(\theta_z|\theta_{i-1})$, which specifies the chance of picking a parameter set θ_z given the current parameter set θ_{i-1} at location $i - 1$ in the Markov chain. Second, the Metropolis-Hastings ratio in equation (12) is computed and used to accept the proposal point with probability α [*Hastings*, 1970]:

$$\alpha = \min\left(1, \frac{\pi(\theta_z)q(\theta_z|\theta_{i-1})}{\pi(\theta_{i-1})q(\theta_{i-1}|\theta_z)}\right). \tag{12}$$

[22] By alternating between these two basic steps, a Markov chain is generated which, under certain regularity conditions, has an invariant distribution, $\pi(\theta)$. In practice, this means that if one looks at values of θ in the Markov chain that are sufficiently far from the starting value, the successively generated parameter combinations will be distributed with stable frequencies stemming from $\pi(\theta)$. For a given choice of the proposal density, the efficiency of an MCMC algorithm, i.e., the number of samples needed to achieve convergence to the posterior density, depends on the problem dimension, i.e., how many parameters are estimated simultaneously, and on the complexity of the posterior distribution. High-dimensional, asymmetric, and multimodal posteriors are generally much more challenging to sample from than low-dimensional, symmetric, and unimodal posteriors. Asymmetry may arise due to strong parameter correlations, yielding elongated ridges of high density in multidimensional parameter space and elongated density contours along two-dimensional projections of $\pi(\theta)$. Both asymmetry and multimodality are in general indicative of parameter nonuniqueness. As shown by *Kavetski et al.* [2006a] and in the next section, multimodality may also be an artifact of poor numerical approximation of $\pi(\theta)$.

[23] In this paper, we use the recently developed DiffeRential Evolution Adaptive Metropolis (DREAM) algorithm [*Vrugt et al.*, 2009] to generate samples from the posterior distribution. This adaptive MCMC scheme evolves multiple interacting chains simultaneously for global exploration and automatically tunes the scale and orientation of the proposal distribution during the search. The implementation used herein uses sampling from an archive of past states. This approach is called DREAM_ZS and has three main advantages, details of which will be given in a later publication. Initial theory and applications can be found in *ter Braak and Vrugt* [2008]. Parameter uncertainty estimation with DREAM_ZS is combined with different numerical methods from section 3 for integrating the water balance model and computing the posterior density $\pi(\theta)$. Numerical integration methods characterized by large errors will produce inaccurate simulations of $\pi(\theta)$, thereby potentially affecting (1) the accuracy of estimated parameter uncertainty and (2) the efficiency of the MCMC algorithm in converging to the posterior density. Ideally, we are looking for numerical integration methods that result in both accurate and efficient estimation of parameter uncertainty.

5. Effect of Numerical Integration Scheme on MCMC Performance

[24] We study effects of the numerical integration method on MCMC performance by means of an example case study. We force the hydrologic model described in section 2 with 3 years of daily rainfall and potential evaporation data from the French Broad basin in North Carolina [*Duan et al.*, 2006]. The likelihood function $f(x|\theta)$ in equation (11) is based on the standard least squares model, i.e., assuming model residuals to be Gaussian, independent, and identically distributed with mean zero and constant variance. The procedure leads to posterior uncertainty estimation of seven model parameters (Table 3). Uniform prior densities $\pi_0(\theta)$ are assumed for all parameters, with prior parameter ranges listed in Table 3. All MCMC trials with DREAM_ZS reported herein consisted of 100,000 function evaluations (model runs) of which 50% was used for burn-in.

[25] We evaluate to what extent numerical implementation of the hydrologic model affects the inferred posterior parameter and predictive distributions, convergence rate of the MCMC algorithm, and computational efficiency of the analysis. Posterior distributions obtained with the integration methods listed in Table 1 are compared to those computed using a numerically accurate implementation of the hydrologic model, based on a fifth-order explicit Runge-Kutta method [*Press et al.*, 1990] with adaptive time stepping using equation (5) and a tolerance τ equal to 10^{-8} (referred to as reference or “REF”). First, we present results on parameter and predictive uncertainty for the three first-order methods (E1, I1, and S1) using fixed daily time stepping. This is fol-

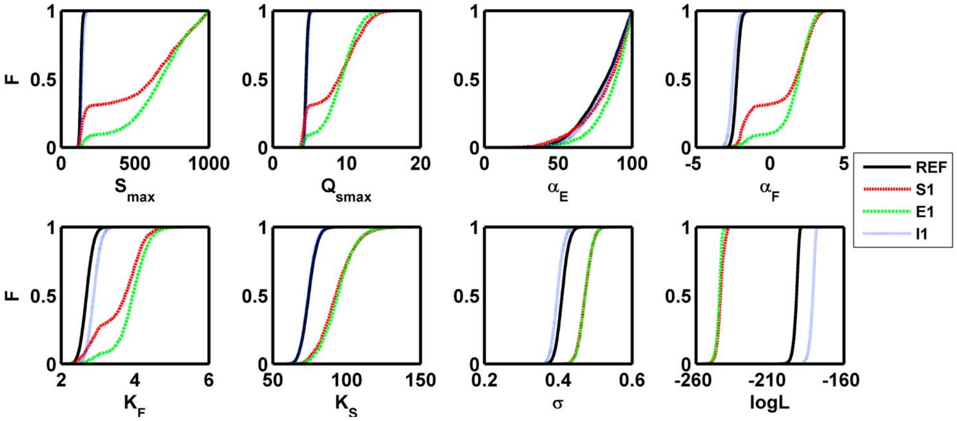

Figure 1. Marginal posterior cumulative distribution functions (cdf's) for hydrologic model parameters using different numerical integration methods. Details on numerical methods are listed in Table 1.

lowed by a comparison to the three second-order methods (E2, I2, S2) using adaptive time stepping with various tolerances.

5.1. First-Order, Fixed-Step Integration Methods: Parameter Uncertainty

[26] Figure 1 shows marginal posterior parameter cumulative distribution functions (cdf's) for all model parameters using different numerical integration methods. Results in Figure 1 show that accurate numerical implementation of the hydrologic model ("REF") yields well-identified parameters with unimodal marginal posterior distributions. On the other hand, the use of first-order, fixed-step explicit integration methods (S1 and E1) yields marginal posterior distributions that deviate significantly from the numerically accurate ones. This is the case for at least four parameters, namely $S_{\max}$, $Q_{s\max}$, α_F, and K_F. Note that three of these parameters are directly connected to the correct simulation of runoff (fast response), suggesting that parameter values are compensating for numerical errors during peak flows. A distinct feature of the marginal posterior cdf's of these parameters is that they display a plateau, especially $S_{\max}$ and α_F, which is indicative of a bimodal distribution. On the other hand, the marginal posterior distribution for K_S, a parameter characterizing baseflow response, is unimodal for all numerical methods, although S1 and E1 show clear deviations. Interestingly, the corresponding distribution of log-likelihood values, also shown in Figure 1, reveals that S1 and E1 have significantly lower log-likelihood values compared to the reference run. The opposite is true for the implicit Euler method (I1): its log-likelihood values are slightly higher than the reference run, confirming findings of *Kavetski et al.* [2006a] and *Kavetski and Clark* [2010]. More importantly, the I1 method does not suffer from the problems encountered with the explicit method, yielding unimodal marginal posterior distributions that match those of the reference run. Again, this confirms the results of *Kavetski and Clark* [2010], although they also show some examples where the fixed-step implicit Euler method results in significantly different posterior parameter distributions, compared to a numerically accurate method.

[27] The results in Figure 1 suggest that the macroscale features of the response surface have been altered when using the explicit numerical integration methods, as indicated by a change from unimodal to bimodal marginal parameter cdf's. Note that this does not necessarily mean that the joint posterior distribution became bimodal [see *Kavetski and Clark*, 2010, Figure 9]. These changes are confirmed in Figure 2, which shows a one-dimensional profile of the log-likelihood response surface along a straight line through the seven-dimensional parameter space, connecting the two marginal posterior modes identified using the S1 method. The correct log-likelihood profile corresponding to the reference run (shown in black) exhibits a clear peak near a value of $S_{\max}$ = 100 mm. Both explicit methods (S1 and E1), on the other hand, have significantly lower log-likelihood values at that point, and exhibit a second peak near $S_{\max}$ = 700 mm, which is however a bit lower than the first peak. This means that if one were only interested in the posterior mode, i.e., the parameter set with the highest log-likelihood value, the explicit methods would in principle yield a numerically accurate answer, at least in this particular case (for counterexamples, see *Clark and Kavetski*, [2010]). However, if the entire posterior distribution is of interest, then

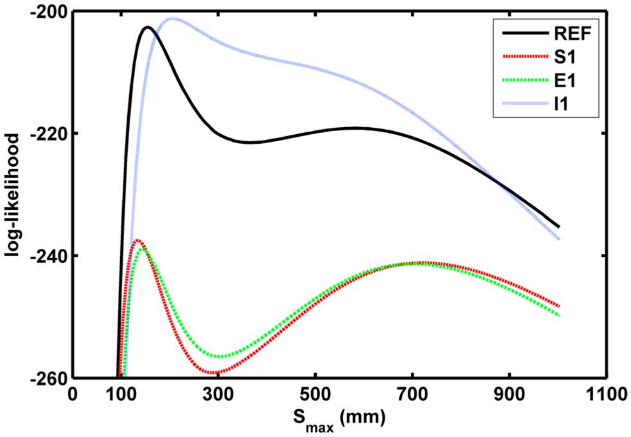

Figure 2. Log-likelihood profile for different numerical methods. The profile is taken along a straight line through seven-dimensional parameter space and is projected on a single parameter axis.

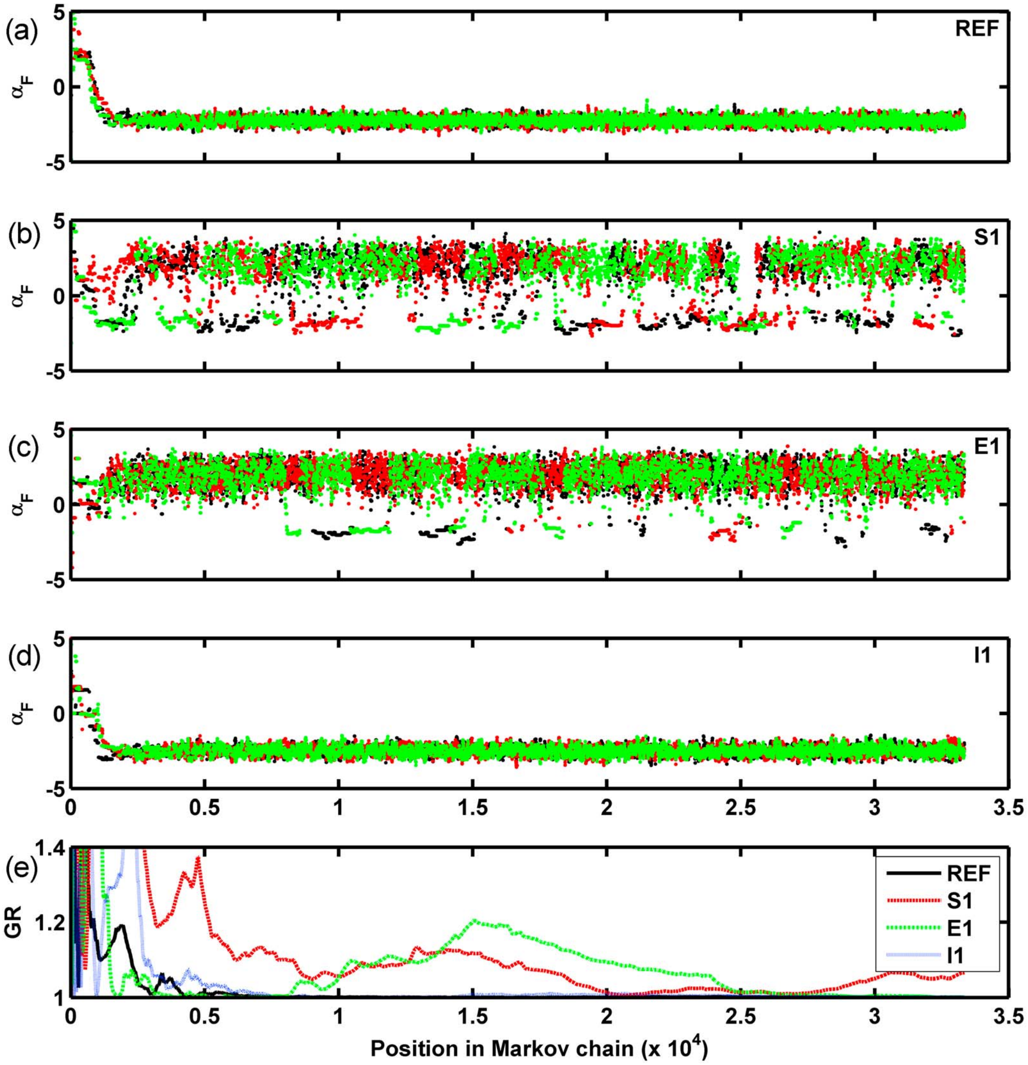

Figure 3. (a–d) Trace plots and (e) Gelman-Rubin convergence statistic for Markov chains of parameter α_F simulated with different numerical integration methods. Each MCMC run consists of three parallel chains with a total of 100,000 model runs.

the second peak near $S_{\max}$ = 700 mm becomes relevant as well. Whether this will affect MCMC sampling of the posterior distribution depends on the ratio of the posterior densities between the two peaks, as indicated by equation (12) and the relative width of both regions of attraction. For example, with the S1 method, there are at least two regions of attraction (Figure 2) with maximum log-likelihood values of respectively −237.7 and −241.2. These two values lie much closer together than the corresponding values for the numerically accurate implementation (Figure 2), where one peak clearly dominates the other. In addition, with the S1 method, the region of attraction associated with the lower log likelihood is much wider than the one associated with the higher log likelihood. This explains why most MCMC samples are drawn from this lower log-likelihood region (Figure 3), as the chance of randomly drawing a parameter set near the peak of the narrow high log-likelihood region is quite small. Indeed, the narrowness of the higher log-likelihood region peak compared to the lower log-likelihood region creates a challenging response surface to sample from.

[28] This is clearly illustrated in Figure 3, which shows trace plots for the parameter α_F, as the MCMC chains evolve toward the posterior distribution. In the reference run, chains mix well, resulting in fast convergence to the limiting distribution. The same behavior is observed for the implicit Euler scheme (I1). Trace plots for the two explicit schemes, however, suggest the existence of two regions of attraction for this parameter, approximately around α_F = −2 and α_F = 2, with the former corresponding to posterior values obtained with the reference run. These results possibly indicate bimodality in the response surface, although it could also be caused by the presence of a narrow, geometrically complex, yet unimodal, ridge. In any case, it is clear that the MCMC algorithm has difficulties converging on this problem, and is hampered by slow mixing of the different Markov chains. In the explicit Euler run (E1), the region around $\alpha_F = -2$ even remains unsampled during the first 20,000 model runs, indicating potential problems with declaring early convergence to the wrong posterior distribution. Figure 3 also plots values for the Gelman-Rubin (GR) statistic [*Gelman and Rubin*,

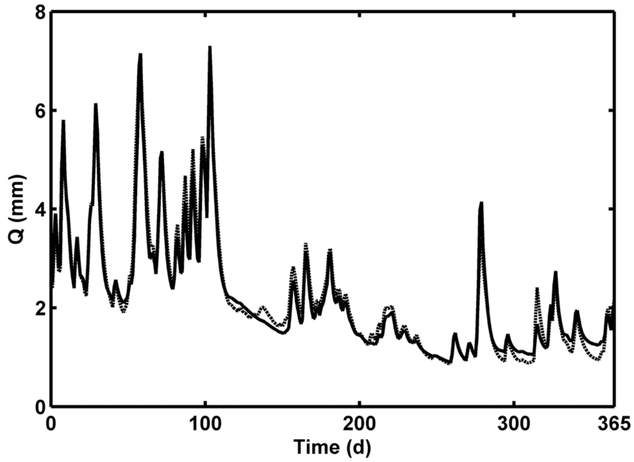

Figure 4. Predicted hydrographs using the S1 method with two different parameter sets, corresponding to the two modes shown in Figure 2.

1992], often used as an indicator of MCMC convergence. This statistic compares parameter mean and variance within and between individual chains, and a value smaller than 1.2 is typically used to declare convergence. Figure 3 suggests that, in the case of complex marginal posteriors for S1 and E1, the GR statistic may not be a reliable convergence indicator, confirming results reported by *Woodward* [2007] (available at http://ftp.isds.duke.edu/pub/WorkingPapers/08-05.html). In addition, results in Figure 3 for method E1 show that based on GR premature convergence would be declared early in the MCMC run, when the numerically accurate mode of the posterior has not been sampled yet. Repeated MCMC runs with the E1 and S1 methods using the convergence criterion GR < 1.05 confirmed this problem: depending on the random initialization and evolution of the Markov chains, the runs either prematurely converged to a marginal posterior distribution centered around $S_{\max}$ = 700 mm, or they did not converge after 100,000 model runs due to poor mixing between individual chains. These results illustrate that numerical implementation of the hydrologic model can have a significant effect on convergence behavior of the MCMC algorithm.

5.2. First-Order, Fixed-Step Integration Methods: Predictive Uncertainty

[29] Next, we studied the effects of inferred parameter uncertainty on predictive streamflow distributions. Figure 4 shows predicted hydrographs using two different parameter sets, namely the ones corresponding to the two posterior modes inferred using the first-order splitting method S1 (one being close to the numerically accurate parameter set, and the other being an artifact of the numerical integration method; see Figure 2). It is evident from this figure that the two parameter sets yield quite similar hydrographs, suggesting that, despite the large variation in parameter sets contained in the posterior distribution inferred with method S1 (Figure 1), predictive uncertainty may not be that large.

[30] Figure 5 shows posterior predictive streamflow distributions due to parameter uncertainty for the different numerical methods at specified times along the hydrograph of Figure 4. Predictive distributions are quite narrow for all methods, but they do not all overlap with the distributions for the reference run. Predictive uncertainty is typically somewhat larger with the explicit methods, due to the wider posterior parameter distributions (Figure 1), but the effect is not that pronounced, although the marginal cdf of the last plot in Figure 5 shows a clear bimodal marginal predictive streamflow distribution for both S1 and E1. Predictions with the explicit methods (S1, E1) typically underestimate streamflow, whereas the implicit Euler method (I1) either underestimates or overestimates. Note that, even though the I1 method outperformed the explicit methods and yielded accurate estimates of parameter uncertainty (Figure 1), its performance in estimating predictive uncertainty is fairly similar to those of the explicit methods (Figure 4).

5.3. Second-Order, Adaptive-Step Integration Methods

[31] Results from the previous section indicate that first-order, fixed-step integration methods may not be appropriate for MCMC analysis of parameter and predictive uncertainty. Here, we wish to investigate whether second-order, adaptive-step integration methods can alleviate these problems. We applied adaptive stepping using second-order methods (S2, E2, and I2) to estimate local numerical errors of the first-order methods as a basis for adjusting the time step, as described in section 3. Hence, we expect to improve accuracy and stability by using adaptive time stepping and a second-order method. Four different tolerances were used in equation (10), namely, $\tau = 10^{-1}$, 10^{-2}, 10^{-3}, and 10^{-4}. We are mainly interested in how parameter estimates, streamflow predictions, and computational speed change as a function of decreasing tolerance. Ideally, we are looking for methods that are both accurate and efficient.

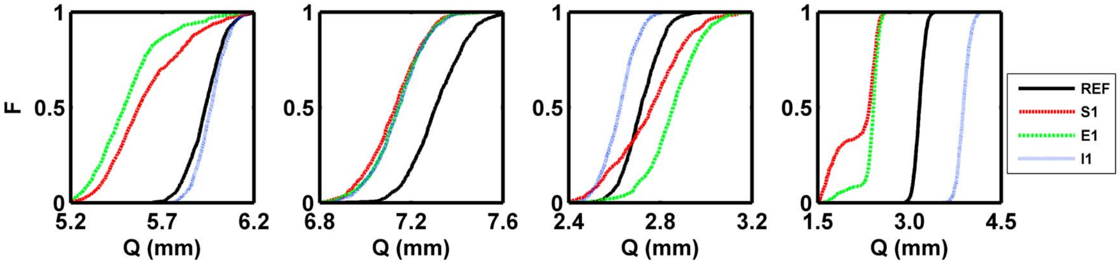

Figure 5. Posterior streamflow cdf's due to parameter uncertainty using different numerical integration methods. Each subplot corresponds to a particular time on the hydrograph in Figure 4; i.e., times are 8, 58, 157, and 315 days, respectively.

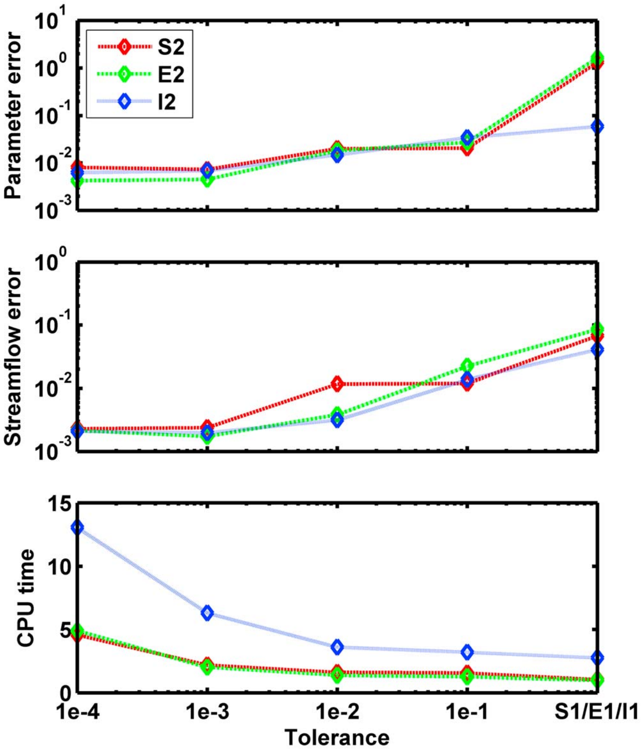

Figure 6. Relation between tolerance in adaptive time-stepping method, computational cost, and accuracy of parameter and streamflow estimates. Parameter and streamflow errors are averages expressed relative to parameter and streamflow values obtained with the reference numerical method (REF). CPU time per model run is relative to CPU times obtained with the E1 method.

[32] Average CPU time for the fastest numerical implementation ranged from 0.0028 s per hydrologic model run (for the fixed-step explicit Euler method) to 0.0176 s per model run (for the adaptive-step, second-order implicit method). Software implementation was based on MCMC code written in MATLAB and the hydrologic model written in FORTRAN. We note that implementation of the hydrologic model in FORTRAN yielded an order-of-magnitude speedup compared to a MATLAB implementation of the same model, due to the necessity of a for loop to account for daily variations in rainfall and potential evaporation forcing when the water balance equations are integrated over time.

[33] Results in Figure 6 clearly show the increase in accuracy with decreasing tolerance, for both parameter estimates and streamflow predictions. It does not take very small tolerances to significantly decrease parameter and streamflow errors (where errors are computed relative to results from the reference run; i.e., a value of 0.1 corresponds to a 10% error relative to the numerically accurate result). A tolerance value of 0.1 already results in order-of-magnitude increases in accuracy, compared to first-order, fixed-step methods, without leading to a significant increase in computational cost. With such a high tolerance value, there are few time step decreases, and therefore the main reason for this significant increase in accuracy is the use of second-order integration methods. Note that the decrease in parameter error is only moderate when going from I1 to I2 with tolerance of 0.1, since I1 already gave quite accurate parameter estimates (see Figure 1). The crucial advantage of I2, however, is its control of numerical errors within a specified tolerance. As tolerances decrease, accuracy increases and computational cost goes up. Based on the results in Figure 6, tolerances around 10^{-3}–10^{-2} seem to give the best trade-off between accuracy and computational speed for this case study with a single-state conceptual model. Note that the explicit finite difference and splitting methods are quite similar in both accuracy and speed. The implicit finite difference method, on the other hand, is at least as accurate, and it is more accurate when it is used with a first-order, fixed-step method, but its computational cost is clearly higher than those of either the explicit or splitting methods. The computational requirements of the implicit methods are larger, since these require iterative solution of a nonlinear problem at each time step. Explicit and splitting methods implemented with small tolerance values give highly accurate solutions at a smaller computational cost. Hence, unless there are other reasons for choosing the implicit method (e.g., microscale smoothness requirements of the response surface for gradient-based optimization, as in the work of *Kavetski et al.* [2006b]), we conclude that adaptive-step explicit and splitting methods work best for Monte Carlo simulation.

6. Discussion

[34] Our results confirm insights by *Kavetski et al.* [2003], who showed that irregular response surfaces in conceptual rainfall-runoff models may be an artifact of inaccurate numerical implementation. In particular, the use of first-order, explicit, fixed-step integration methods is problematic, as it results in numerically inaccurate parameter uncertainty estimates (Figure 1). Although numerical errors altered the shape of the response surface, they did not significantly affect the location of the globally optimal parameter set (Figure 2). However, the presence of multiple local optima would make any optimization unnecessarily challenging. As a test, we applied a global optimization algorithm (SCE-UA of *Duan et al.* [1993]) to our problem with the aim of finding the global optimum (located near $S_{\max}$ = 100 mm in Figure 2) when integrating the hydrologic model with the S1 method. The algorithm invariably converged toward the secondary mode located near $S_{\max}$ = 700 mm. The particular shape of the response surface in this case, i.e., a narrow peak near $S_{\max}$ = 100 mm, combined with a much broader (but inferior) optimum near $S_{\max}$ = 700 mm, apparently creates a very challenging optimization problem. This underscores the need for accurate numerical implementation of the hydrologic model, even when interest is only in the optimal parameter set as opposed to the entire posterior distribution.

[35] Problems with the Gelman-Rubin statistic in Figure 3 highlight the need for alternative convergence diagnostics [e.g., *Cowles and Carlin*, 1996] when posterior distributions deviate significantly from the normal case. Indeed, multimodality of the marginal or joint posterior parameter distributions may not be restricted to poorly implemented models, but can also occur in cases with numerically accurate models, e.g., when overparameterized models are used. Therefore, it is recommended to always evaluate the robust-

ness of inferred parameter distributions, either by repeating the MCMC analysis from different starting points or by extending the length of the Markov chains to avoid premature convergence to a unimodal distribution.

[36] Convergence problems due to inaccurate numerical implementation, as depicted in Figure 3, are likely to become more severe for more complex models, with additional nonlinear reservoirs and parameters. When the model becomes overparameterized, the model parameter response surface may become more complex, potentially with multiple optima (ill-posedness); hence the inferred posterior distributions may be especially sensitive to slight changes in the relative magnitude of different peaks caused by numerical integration errors. The size of the numerical errors will also depend on the inherent flow dynamics of the basin. Using fixed-step explicit methods to simulate highly dynamic processes is bound to result in greater errors. For example, in Figure 2 it can be seen that numerical errors, as measured by differences between log-likelihood profiles for S1 and REF methods, are greater for smaller values of $S_{\max}$, as smaller storage capacities result in more dynamic changes in the relative water storage, S_r.

[37] Which numerical integration method to use as part of an MCMC analysis is determined by the trade-off between accuracy and computational cost (Figure 6). In our case the adaptive time stepping with either a second-order explicit scheme (E2) or a second-order splitting method (S2) worked best. These methods resulted in efficient and accurate estimates of parameter and prediction uncertainty, even with relatively large tolerances. The E2 method has the additional advantage of not depending on analytical tractability of the hydrologic flux laws, as is the case for the S2 method. The latter also requires careful numerical implementation (see the appendix), and may be harder to expand to model structures with multiple interacting reservoirs with feedback. The fixed-step, implicit Euler method performed well, too, although it is computationally more costly due to its reliance on nonlinear iterations at each time step. It may be preferred for gradient-based optimization of rainfall-runoff models, as it guarantees smoothness of the parameter response surface, while providing accurate results [*Kavetski et al.*, 2006b]. Monte Carlo methods, however, do not rely on continuous derivatives of the response surface, hence microscale smoothness is less of an issue, allowing one to use more efficient adaptive-step methods that preserve the macroscale features of the response surface.

[38] Finally, our results can be compared to a two-part recent study by *Clark and Kavetski* [2010] and *Kavetski and Clark* [2010]. They investigated the effects of various numerical integration methods on model prediction, parameter sensitivity, inference, and optimization using a range of hydrologic basins and various hydrologic model structures. In terms of the effects of inaccurate numerical integration using fixed-step explicit methods on MCMC simulation, *Kavetski and Clark* [2010] show that MCMC-based posterior parameter distributions inferred with fixed-step explicit methods may deviate significantly from those obtained with numerically accurate methods. Our results for E1 and S1 in Figure 1 confirm these findings. On the other hand, we find that the fixed-step implicit Euler method yields quite accurate parameter uncertainty estimates (see I1 in Figure 1), whereas *Kavetski and Clark* [2010] conclude that this is not always the case. When the I1 method is used, the log-likelihood profile is clearly affected (Figure 3), but whether this has significant repercussions on inferred parameter uncertainty apparently is case-specific. Both implicit and explicit fixed-step methods introduce inaccuracies in streamflow predictions (Figure 5). Adaptive time stepping methods are very effective for avoiding such problems, as shown here and also by *Kavetski and Clark* [2010], who also recommend the use of adaptive-step, second-order explicit finite differencing.

[39] The general similarities of their results, which were based on multiple model structures and multiple basins, validate our results for a single basin and a single model structure, and suggest that the conclusions reached in this paper have more general applicability to conceptual hydrologic models. Furthermore, as the setup in this paper can be considered an optimistic scenario (i.e., use of a state-of-the-art MCMC algorithm combined with a simple spatially lumped hydrologic model), effects on accuracy and efficiency observed in this case will likely be amplified when less powerful MCMC algorithms are used in combination with more complex conceptual hydrologic models.

7. Conclusions

[40] Our results show that accuracy and efficiency of posterior parameter uncertainty estimation using MCMC sampling not only depends on the design of the MCMC algorithm itself, but also depends on the numerical scheme used to integrate the underlying hydrologic model equations. Using a conceptual rainfall-runoff model as a case study, we showed that numerical schemes based on fixed-step explicit time integration, representative of many ad hoc implementations of conceptual rainfall-runoff models, dramatically changed the macroscale features of the multidimensional parameter response surface, confirming results first reported by *Kavetski et al.* [2003]. This resulted in numerical artifacts, such as bimodal marginal posterior parameter distributions, that were not present in numerically accurate implementations of the same hydrologic model. Consequences for MCMC performance include (1) numerically inaccurate estimates of posterior parameter and predictive distributions, (2) poor performance and slow convergence of the MCMC algorithm, and (3) unreliable convergence diagnosis using the Gelman-Rubin statistic. The fact that we observed these issues on a relatively small parameter inference problem using a state-of-the-art MCMC algorithm reinforces the necessity of a proper, accurate numerical implementation of hydrologic models.

[41] A fixed-step implicit Euler scheme avoided these numerical artifacts and resulted in numerically accurate estimates of parameter and predictive uncertainty. However, as this scheme does not include any error control, it does not protect against potential problems in other cases. A more robust alternative is provided by adaptive-step, second-order integration methods, which were all shown here to remedy numerical problems with fixed-step explicit methods. With adaptive-step methods, the desired accuracy is prescribed by the user-specified tolerance. Therefore, computational speed becomes the main criterion for choosing between different adaptive-step methods. Our results show that adaptive-step, second-order methods, based on either explicit finite differencing or operator splitting with analytical integration,

provide the best alternative for accurate and efficient MCMC simulation. Explicit finite differencing is preferred if the individual hydrologic flux laws cannot be integrated analytically or if the model structure consists of multiple, coupled, nonlinear reservoirs, as the splitting method then loses its advantage. Implicit methods may also be used for increased accuracy, but they cannot match the efficiency of adaptive-step explicit finite differencing or operator splitting for integrating spatially lumped conceptual hydrologic models.

Appendix

[42] This appendix shows how the analytical solutions in Table 2 were implemented to minimize cancellation errors due to the summation of exponentials of widely varying magnitude. The solutions in Table 2 can be written in the following general form:

$$S_r = \frac{1}{\alpha}\log(e^{v_1} - e^{v_2} + e^{v_3}), \qquad \text{(A1)}$$

where v_1, v_2, and v_3 take on different values depending on the hydrologic process (Table 2). Poor scaling of the sum of exponentials is minimized by rewriting (A1) as

$$S_r = \frac{1}{\alpha}[c + \log(e^{v_1-c} - e^{v_2-c} + e^{v_3-c})] \qquad \text{(A2)}$$

and by setting c equal to the largest v_i (in absolute value), with i = 1, 2, 3. Without this procedure, numerical implementation of the analytical solutions resulted in non-negligible numerical errors.

[43] **Acknowledgments.** We would like to thank Dmitri Kavetski and two anonymous reviewers for constructive reviews that significantly improved a previous version of this paper. The second author was supported by a J. Robert Oppenheimer Fellowship from the LANL postdoctoral program.

References

Bates, B. C., and E. P. Campbell (2001), A Markov chain Monte Carlo scheme for parameter estimation and inference in conceptual rainfall-runoff modeling, *Water Resour. Res.*, *37*(4), 937–947.

Butcher, J. C. (2008), *Numerical Methods for Ordinary Differential Equations*, John Wiley, Chichester, U. K.

Clark, M. P., and D. Kavetski (2010), The ancient numerical daemons of conceptual hydrological modeling: Fidelity and efficiency of time stepping schemes, *Water Resour. Res.*, W10510, doi:10.1029/2009WR008894.

Cowles, M. K., and B. P. Carlin (1996), Markov chain Monte Carlo convergence diagnostics: A comparative review, *J. Am. Stat. Assoc.*, *91*(434), 883–904.

Cox, B. A., and P. G. Whitehead (2005), Parameter sensitivity and predictive uncertainty in a new water quality model Q2, *J. Environ. Eng.*, *131*(1), 147–157.

Csomos, P., and I. Farago (2008), Error analysis of the numerical solution of split differential equations, *Math. Comput. Modell.*, *48*, 1090–1106.

Csomos, P., I. Farago, and A. Havasi (2005), Weighted Sequential Splittings and Their Analysis, *Comput. Math. Appl.*, *50*, 1017–1031.

Duan, Q., V. K. Gupta, and S. Sorooshian (1993), Shuffled complex evolution approach for effective and efficient global minimization, *J. Optim. Theory Appl.*, *76*(3), 501–521.

Duan, Q., et al. (2006), Model Parameter Estimation Experiment (MOPEX): An overview of science strategy and major results from the second and third workshops, *J. Hydrol.*, *320*(1–2), 3–17.

Engeland, K., C. Y. Xu, and L. Gottschalk (2005), Assessing uncertainties in a conceptual water balance model using Bayesian methodology, *Hydrol. Sci. J.*, *50*(1), 45–63.

Fenicia, F., H. H. G. Savenije, P. Matgen, and L. Pfister (2007), A comparison of alternative multiobjective calibration strategies for hydrological modeling, *Water Resour. Res.*, *43*, W03434, doi:10.1029/2006WR005098.

Gelman, A., and D. B. Rubin (1992), Inference from iterative simulation using multiple sequences, *Stat. Sci.*, *7*, 457–472.

Hastings, W. K. (1970), Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, *57*, 97–109.

Kavetski, D., and M. P. Clark (2010), The ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction, *Water Resour. Res.*, W10511, doi:10.1029/2009WR008896.

Kavetski, D., and G. Kuczera (2007), Model smoothing strategies to remove microscale discontinuities and spurious secondary optima in objective functions in hydrological calibration, *Water Resour. Res.*, *43*, W03411, doi:10.1029/2006WR005195.

Kavetski, D., P. Binning, and S. W. Sloan (2001), Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards equation, *Adv. Water Resour.*, *24*(6), 595–605.

Kavetski, D., G. Kuczera, and S. W. Franks (2003), Semidistributed hydrological modeling: A "saturation path" perspective on TOPMODEL and VIC, *Water Resour. Res.*, *39*(9), 1246, doi:10.1029/2003WR002122.

Kavetski, D., G. Kuczera, and S. W. Franks (2006a), Calibration of conceptual hydrological models revisited: 1. Overcoming numerical artefacts, *J. Hydrol.*, *320*(1–2), 173–186.

Kavetski, D., G. Kuczera, and S. W. Franks (2006b), Calibration of conceptual hydrological models revisited: 2. Improving optimization and analysis, *J. Hydrol.*, *320*(1–2), 187–201.

Kuczera, G. (1983), Improved parameter inference in catchment models: 1. Evaluating parameter uncertainty, *Water Resour. Res.*, *19*(5), 1151–1162.

Kuczera, G., and E. Parent (1998), Monte Carlo assessment of parameter uncertainty in conceptual catchment models: The Metropolis algorithm, *J. Hydrol.*, *211*(1–4), 69–85.

Kuczera, G., D. Kavetski, B. Renard, and M. Thyer (2010), A limited-memory acceleration strategy for MCMC sampling in hierarchical Bayesian calibration of hydrological models, *Water Resour. Res.*, *46*, W07602, doi:10.1029/2009WR008985.

McLachlan, R. I., and G. R. W. Quispel (2002), Splitting methods, *Acta Numer.*, *11*, 341–434, doi:10.1017/S0962492902000053.

Parkhurst, D. L., and C. A. J. Appelo (1999), User's guide to PHREEQC—A computer program for speciation, reaction-path, 1D-transport, and inverse geochemical calculations, *U.S. Geol. Surv. Water Resour. Invest. Rep.*, *99-4259*, 312 pp.

Perrin, C., C. Michel, and V. Andreassian (2003), Improvement of a parsimonious model for streamflow simulation, *J. Hydrol.*, *279*(1–4), 275–289.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1990), *Numerical Recipes*, Cambridge Univ. Press, New York.

Qu, Y., and C. J. Duffy (2007), A semi-discrete finite volume formulation for multiprocess watershed simulation, *Water Resour. Res.*, *43*, W08419, doi:10.1029/2006WR005752.

Robert, C. P., and G. Casella (2004), *Monte Carlo Statistical Methods*, 2nd ed., 536 pp., Springer, New York.

Singh, V. P., and D. A. Woolhiser (2002), Mathematical modeling of watershed hydrology, *J. Hydrol. Eng.*, *7*(4), 270–292.

Smith, T. J., and L. A. Marshall (2008), Bayesian methods in hydrologic modeling: A study of recent advancements in Markov chain Monte Carlo techniques, *Water Resour. Res.*, *44*, W00B05, doi:10.1029/2007WR006705.

Steefel, C. I., and K. T. B. MacQuarrie (1996), Approaches to modeling of reactive transport in porous media, *Rev. Mineral.*, *34*, 83–129.

Strang, G. (1968), On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.*, *5*(3), 506–517.

ter Braak, C. J. F., and J. A. Vrugt (2008), Differential evolution Markov chain with snooker updater and fewer chains, *Stat. Comput.*, *18*(4), 435–446.

Vrugt, J. A., H. V. Gupta, W. Bouten, and S. Sorooshian (2003), A shuffled complex evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, *Water Resour. Res.*, *39*(8), 1201, doi:10.1029/2002WR001642.

Vrugt, J. A., H. V. Gupta, S. C. Dekker, S. Sorooshian, T. Wagener, and W. Bouten (2006), Application of stochastic parameter optimization to the Sacramento soil moisture accounting model, *J. Hydrol.*, *325*(1–4), 288–307.

Vrugt, J. A., C. J. F. ter Braak, M. P. Clark, J. M. Hyman, and B. A. Robinson (2008), Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation, *Water Resour. Res.*, *44*, W00B09, doi:10.1029/2007WR006720.

Vrugt, J. A., C. J. F. ter Braak, C. G. H. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon (2009), Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling, *Int. J. Nonlinear Sci. Numer. Simul.*, *10*(3), 273–290.

Wagener, T., M. Sivapalan, P. Troch, and R. Woods (2007), Catchment classification and hydrologic similarity, *Geogr. Compass*, *1*(4), 901–931.

Wood, E. F., D. P. Lettenmaier, and V. G. Zartarian (1992), A land-surface hydrology parameterization with subgrid variability for general circulation models, *J. Geophys. Res.*, *97*(D3), 2717–2728.

Woodward, D. B. (2007), Detecting poor convergence of posterior samplers due to multimodality, *Discuss. Pap. 2008-05*, Dept. of Stat. Sci. Duke Univ., Durham, N. C.

F. Fenicia, G. Schoups, and N. C. van de Giesen, Department of Water Management, Delft University of Technology, Stevinweg 1, P.O. Box 5048, 2600 GA Delft, Netherlands. (g.h.w.schoups@tudelft.nl)

J. A. Vrugt, Department of Civil and Environmental Engineering, University of California, Irvine, CA 92697, USA.