High-dimensional posterior exploration of hydrologic models using multiple-try DREAM$_{(ZS)}$ and high-performance computing

Eric Laloy[1] and Jasper A. Vrugt[1,2]

Received 26 February 2011; revised 7 October 2011; accepted 6 December 2011; published 20 January 2012.

[1] Spatially distributed hydrologic models are increasingly being used to study and predict soil moisture flow, groundwater recharge, surface runoff, and river discharge. The usefulness and applicability of such complex models is increasingly held back by the potentially many hundreds (thousands) of parameters that require calibration against some historical record of data. The current generation of search and optimization algorithms is typically not powerful enough to deal with a very large number of variables and summarize parameter and predictive uncertainty. We have previously presented a general-purpose Markov chain Monte Carlo (MCMC) algorithm for Bayesian inference of the posterior probability density function of hydrologic model parameters. This method, entitled differential evolution adaptive Metropolis (DREAM), runs multiple different Markov chains in parallel and uses a discrete proposal distribution to evolve the sampler to the posterior distribution. The DREAM approach maintains detailed balance and shows excellent performance on complex, multimodal search problems. Here we present our latest algorithmic developments and introduce MT-DREAM$_{(ZS)}$, which combines the strengths of multiple-try sampling, snooker updating, and sampling from an archive of past states. This new code is especially designed to solve high-dimensional search problems and receives particularly spectacular performance improvement over other adaptive MCMC approaches when using distributed computing. Four different case studies with increasing dimensionality up to 241 parameters are used to illustrate the advantages of MT-DREAM$_{(ZS)}$.

Citation: Laloy, E., and J. A. Vrugt (2012), High-dimensional posterior exploration of hydrologic models using multiple-try DREAM$_{(ZS)}$ and high-performance computing, *Water Resour. Res.*, *48*, W01526, doi:10.1029/2011WR010608.

1. Introduction and Scope

[2] The ever increasing pace of computational power, along with significant advances in measurement technologies, and interests in real-time forecasting has stimulated the development of increasingly complex spatially distributed hydrologic models. The usefulness and applicability of these models depends strongly on the values of the model parameters. Unfortunately, the estimation of the correct values of these parameters has not proved to be simple. To cope with the issues of heterogeneities, scale effects, and process complexity, many models use effective parameters to aggregate complex interactions driven by a number of highly interrelated energy and mass transport processes. The consequence of this process aggregation is that the parameters cannot be directly measured in the field but can only be meaningfully inferred by adjusting them so that the behavior of the model approximates, as closely and consistently as possible, the observed system behavior over some historical period of time. Sparse data and regionalization relationships may be used to constrain the model by reducing the number of parameters, but the resulting inverse problem nevertheless involves iterative improvement through successive executions of the model, a situation that places a premium on calibration methods that can efficiently summarize parameter and predictive uncertainty.

[3] The past two decades have seen an increasing interest in Markov chain Monte Carlo methods for calibration of hydrologic models, and treatment of parameter, model structural, forcing data, and calibration data uncertainty [see, e.g., *Vrugt et al.*, 2003, 2008; *Keating et al.*, 2010; *Schoups and Vrugt*, 2010]. The basis of this method is a Markov chain that generates a random walk through the search space and iteratively finds solutions with stable frequencies stemming from a fixed probability distribution. To visit configurations with a stable frequency, an MCMC algorithm generates trial moves from the current position of the Markov chain at time $t-1$, $\mathbf{x}_{t-1}$, to a new state $\mathbf{z}$. The earliest MCMC approach is perhaps the well known random walk Metropolis (RWM) algorithm [*Metropolis et al.*, 1953]. Assuming that a random walk has already sampled the points $\{\mathbf{x}_0, \ldots, \mathbf{x}_{t-1}\}$, this algorithm proceeds in the following three steps. First, a candidate point $\mathbf{z}$ is sampled from a proposal distribution $q(\cdot)$ that is symmetric, $q(\mathbf{x}_{t-1}, \mathbf{z}) = q(\mathbf{z}, \mathbf{x}_{t-1})$ and may depend on the present

[1]Department of Civil and Environmental Engineering, University of California, Irvine, California, USA.
[2]Institute for Biodiversity and Ecosystems Dynamics, University of Amsterdam, Amsterdam, Netherlands.

0043-1397/12/2011WR010608

location, $\mathbf{x}_{t-1}$. Next, the candidate point is either accepted or rejected using the Metropolis acceptance probability:

$$\alpha(\mathbf{x}_{t-1}, \mathbf{z}) = \begin{cases} \min\left[\dfrac{\pi(\mathbf{z})}{\pi(\mathbf{x}_{t-1})}, 1\right] & \text{if } \pi(\mathbf{x}_{t-1}) > 0 \\ 1 & \text{if } \pi(\mathbf{x}_{t-1}) = 0 \end{cases} \quad (1)$$

where $\pi(\cdot)$ denotes the density of the target distribution. Finally, if the proposal is accepted, the chain moves to $\mathbf{z}$; otherwise the chain remains at its current location $\mathbf{x}_{t-1}$.

[4] The original RWM scheme was constructed to maintain detailed balance with respect to $\pi(\cdot)$ at each step in the chain:

$$\pi(\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \rightarrow \mathbf{z}) = \pi(\mathbf{z}) p(\mathbf{z} \rightarrow \mathbf{x}_{t-1}) \quad (2)$$

where $\pi(\mathbf{x}_{t-1})$ $(\pi(\mathbf{z}))$ denotes the probability of finding the system in state $\mathbf{x}_{t-1}(\mathbf{z})$, and $p(\mathbf{x}_{t-1} \rightarrow \mathbf{z})(p(\mathbf{z} \rightarrow \mathbf{x}_{t-1}))$ denotes the conditional probability of performing a trial move from $\mathbf{x}_{t-1}$ to $\mathbf{z}(\mathbf{z}$ to $\mathbf{x}_{t-1})$. The result is a Markov chain which, under certain regularity conditions, has a unique stationary distribution with probability density function (pdf) $\pi(\cdot)$. In practice, this means that if one looks at the values of $\mathbf{x}$ generated by the RWM that are sufficiently far from the starting value, the successively generated parameter combinations will be distributed with stable frequencies stemming from the underlying posterior pdf of $\mathbf{x}$, $\pi(\cdot)$. *Hastings* [1970] extended equation (2) to include nonsymmetrical proposal distributions, i.e., $q(\mathbf{x}_{t-1}, \mathbf{z}) \neq q(\mathbf{z}, \mathbf{x}_{t-1})$, in which a proposal jump to $\mathbf{z}$ and the reverse jump do not have equal probability. This extension is called the Metropolis-Hastings algorithm (MH), and has become the basic building block of many existing MCMC sampling schemes.

[5] Existing theory and experiments prove convergence of well-constructed MCMC schemes to the appropriate limiting distribution under a variety of different conditions. Yet, in practice the convergence rate is often disturbingly slow. This inefficiency is typically caused by an inappropriate selection of the orientation and scale of the proposal distribution, $q(\mathbf{x}_{t-1}, \cdot)$, used to generate transitions in the Markov chain. When the proposal distribution is too wide, very few candidate points will be accepted, and the chain will not mix properly and converge rather slowly to the posterior target distribution. On the other hand, if the proposal distribution is too narrow, the chain will remain in close vicinity of its current location, and it will require a very large number of iterations before the entire posterior distribution has been sampled. The selection of the proposal distribution is therefore crucial in determining the efficiency and practical applicability of MCMC simulation.

[6] In the past decade, a variety of different approaches have been proposed to increase the efficiency of MCMC simulation and enhance the original RWM and MH algorithms. These approaches can be grouped into single- and multiple-chain methods. Single-chain methods work with a single trajectory, and continuously adapt the covariance, $\mathbf{\Sigma}$, of a Gaussian proposal distribution, $q(\mathbf{x}_{t-1}, \cdot) = N_d(\mathbf{x}_{t-1}, s_d \mathbf{\Sigma})$, using the information contained in the sample path of the chain, $\mathbf{\Sigma} = \mathrm{Cov}(\mathbf{x}_0, \ldots, \mathbf{x}_{t-1}) + \varepsilon \mathbf{I}_d$. The variable s_d represents a scaling factor (scalar) that depends only on the dimensionality d of the problem, $\mathbf{x}$ is a d-dimensional vector, $\mathbf{I}_d$ signifies the d-dimensional identity matrix, and ε is a small scalar that slightly inflates the actual covariance, $\mathbf{\Sigma}$, so that the entire parameter space can theoretically be sampled. As a basic choice, the scaling factor is chosen to be $s_d = 2.4^2/d$ which is optimal for Gaussian target and proposal distributions [*Roberts et al.*, 1997], and $\varepsilon = 10^{-6}$. Examples of self-adaptive single-chain methods include the adaptive Metropolis (AM) [*Haario et al.*, 2001] and delayed rejection adaptive Metropolis (DRAM) algorithms [*Haario et al.*, 2006]. Component-wise updating of $\mathbf{x}$ [*Haario et al.*, 2005] is possible to increase efficiency of AM for high-dimensional problems (large d). In addition, for the special case of hierarchical Bayesian inference of hydrologic models, *Kuczera et al.* [2010] recently proposed to tune $\mathbf{\Sigma}$ using a limited-memory multiblock presampling step, prior to a classical single block Metropolis run.

[7] Multiple-chain methods use different trajectories running in parallel to explore the posterior target distribution. The use of multiple chains has several desirable advantages, particularly when dealing with complex posterior distributions involving long tails, correlated parameters, multimodality, and numerous local optima [*Gilks et al.*, 1994; *Liu et al.*, 2000; *ter Braak*, 2006; *ter Braak and Vrugt*, 2008; *Vrugt et al.*, 2009; *Craiu et al.*, 2009]. The use of multiple chains offers a robust protection against premature convergence, and opens up the use of a wide array of statistical measures to test whether convergence to a limiting distribution has been achieved [*Gelman and Rubin*, 1992]. One popular multichain method that has found widespread application and use in hydrology is the shuffled complex evolution Metropolis algorithm (SCEM-UA) [*Vrugt et al.* 2003]. Numerical experiments on a diverse set of mathematical test functions have shown that SCEM-UA works well in practice. Yet SCEM-UA does not generate a perfectly reversible Markov chain. The explicit removal of outlier trajectories and covariance updating step violate detailed balance. This poses questions on whether SCEM-UA generates an exact sample of the posterior distribution. With some simple modifications, SCEM-UA could be made an exact sampler, but this is beyond the scope of the current paper. We therefore consider the more recent Differential Evolution Markov Chain (DE-MC) method of *ter Braak* [2006]. This method is relatively easy to illustrate and understand, and can be coded in just a few lines. DE-MC uses differential evolution as genetic algorithm for population evolution with a Metropolis selection rule to decide whether candidate points should replace their parents or not.

[8] In DE-MC, N different Markov chains are run simultaneously in parallel. If the state of a single chain is given by a single d-dimensional vector $\mathbf{x}$, then at each generation the N chains in DE-MC define a population $\mathbf{X}$, which corresponds to an $N \times d$ matrix, with each chain as a row. Jumps in each chain $i = \{1, \ldots, N\}$ are generated by taking a fixed multiple of the difference of two randomly chosen members (chains) of $\mathbf{X}$ (without replacement) with indexes r_1 and r_2:

$$\mathbf{z}^i = \mathbf{x}^i_{t-1} + \gamma(\mathbf{X}^{r_1}_{t-1} - \mathbf{X}^{r_2}_{t-1}) + \boldsymbol{\epsilon}, \qquad r1 \neq r2 \neq i \quad (3)$$

where γ is a user-defined scalar, and $\boldsymbol{\epsilon}$ is drawn from a symmetric d-dimensional distribution with a small variance

compared to that of the posterior, but with unbounded support. The difference vector in equation (3) contains the desired information about the scale and orientation of the target distribution, $\pi(\mathbf{x}|\cdot)$. By accepting each jump with the Metropolis ratio, $\alpha(\mathbf{x}_{t-1}, \mathbf{z}) = \min[\pi(\mathbf{z}^i|\cdot)/\pi(\mathbf{x}^i_{t-1}|\cdot), 1]$, a Markov chain is obtained, the stationary or limiting distribution of which is the posterior distribution. The proof of this is given in *ter Braak and Vrugt* [2008] and *Vrugt et al.* [2008, 2009]. Because the joint pdf of the N chains factorizes to $\pi(\mathbf{x}^1|\cdot) \times \ldots \times \pi(\mathbf{x}^N|\cdot)$, the states $\mathbf{x}^1 \ldots \mathbf{x}^N$ of the individual chains are independent at any generation after DE-MC has become independent of its initial value. After this burn-in period, the convergence of a DE-MC run can thus be monitored with the $\hat{R}$ statistic of *Gelman and Rubin* [1992]. From the guidelines of s_d in random walk Metropolis, the optimal choice of γ is $2.4/\sqrt{2d}$. Every 10^{th} generation, $\gamma = 1.0$ to facilitate jumping between different modes [*ter Braak*, 2006].

[9] DE-MC solves an important practical problem in random walk Metropolis, namely, that of choosing an appropriate scale and orientation for the jumping distribution. Earlier approaches such as (parallel) adaptive direction sampling [*Gilks et al.*, 1994; *Roberts and Gilks*, 1994; *Gilks and Roberts*, 1996] solved the orientation problem but not the scale problem. Vrugt and coworkers [*Vrugt et al.*, 2008, 2009] showed that the efficiency of DE-MC can be enhanced, sometimes dramatically, using self-adaptive randomized subspace sampling and explicit consideration of aberrant trajectories. This method, entitled differential evolution adaptive Metropolis (DREAM), maintains detailed balance and ergodicity and has shown to exhibit excellent performance on a wide range of model calibration studies [e.g., *Vrugt et al.*, 2008; *Dekker et al.*, 2012; *Laloy et al.*, 2010a, 2010b; *Scharnagl et al.*, 2010].

[10] Unfortunately, standard DREAM (DE-MC) requires at least $N = d/2$ to d ($N = 2d$) chains to be run in parallel. Running many parallel chains is a potential source of inefficiency, as each individual chain requires burn-in to travel to the posterior distribution. The lower the number of chains required, the greater the practical applicability of DREAM for computationally demanding posterior exploration problems. One device that enables using a smaller N is to generate jumps in equation (3) from past states of the different chains. *ter Braak and Vrugt* [2008] incorporated this idea into DE-MC and showed by numerical simulation and real-world examples that this method works well up to $d = 100$ using only $N = 3$ chains. These findings inspired Vrugt et al. (J. Vrugt et al., Posterior exploration using differential evolution adaptive Metropolis with sampling from past states, manuscript in preparation, 2012) to create DREAM$_{(ZS)}$, which capitalizes on the advantages of DREAM for posterior exploration but generates candidate points in each individual Markov chain by sampling from an archive of past states. This has several practical and theoretical advantages. Most importantly, only a few parallel chains ($N = 3 - 5$) are required for posterior sampling. This reduces burn-in, particularly for problems involving many parameters (large d), thereby increasing sampling efficiency. Indeed, initial studies to date presented by Vrugt et al. (manuscript in preparation, 2012) have shown that DREAM$_{(ZS)}$ requires fewer function evaluations than DREAM to converge to the appropriate limiting distribution. In DREAM$_{(ZS)}$, the states of the chains are periodically stored in an archive using a simple thinning rule. The size of this matrix steadily increases during sampling, but the relative growth decreases linearly with generation number. This diminishing adaptation of the transition kernel ensures convergence of the individual chains to the posterior distribution [*Roberts and Rosenthal*, 2007]. To increase the diversity of the proposals, DREAM$_{(ZS)}$ additionally includes a snooker updater with adaptive step size. The snooker axis runs through the states of two different chains, and the orientation of this jump is different from the parallel direction update utilized in DREAM. The algorithmic implementation of the snooker update within the context of DE-MC is described by *ter Braak and Vrugt* [2008].

[11] Despite significant enhancements in the efficiency of MCMC methods, it remains typically difficult to solve very high-dimensional posterior exploration problems involving hundreds or thousands of parameters. The performance of optimization and search methods typically deteriorates exponentially with increasing dimensionality of the parameter space. In applied mathematics, this phenomenon is also referred to as the curse of dimensionality. This term was coined by Richard E. Bellman, within the context of dynamic programming. In this paper, we present a general framework for efficient inversion of highly parameterized models. This method, entitled MT-DREAM$_{(ZS)}$, combines the strengths of differential evolution, subspace exploration, sampling from past states, snooker updating, and multiple-try Metropolis sampling [*Liu et al.*, 2000] to efficiently explore high-dimensional posterior distributions. This novel approach maintains detailed balance and ergodicity and takes maximum advantage of distributed computing resources. Four case studies with increasing complexity are used to demonstrate the advantages of MT-DREAM$_{(ZS)}$ over current state-of-the-art optimization and MCMC algorithms, including the parameter estimation toolbox (PEST) [*Doherty*, 2009], the shuffled complexes with principal component analysis (SP-UCI) [*Chu et al.*, 2010], DREAM [*Vrugt et al.*, 2009], and DREAM$_{(ZS)}$ (Vrugt et al., manuscript in preparation, 2012).

[12] This paper is organized as follows. Section 2 presents the key concepts of MT-DREAM$_{(ZS)}$ and discusses how to incorporate this new MCMC method on a high-performance computing platform. In section 3, we evaluate our algorithm against other state-of-the-art MCMC methods, for two known mathematical benchmark distributions involving multimodality and high dimensionality. This is followed by a real-world case study consisting of the calibration of the Sacramento soil moisture accounting model (SAC-SMA) using daily discharge data from the Leaf River in Mississippi. This study involves only 13 parameters, but illustrates the severity of the hydrologic model calibration problem. We conclude our testing of MT-DREAM$_{(ZS)}$ with a CPU-efficient 241-parameter groundwater model. This model calibration problem has been described in detail by *Keating et al.* [2010] and is used to illustrate the superior search capabilities of MT-DREAM$_{(ZS)}$. Finally, section 4 draws conclusions about the presented work and discusses yet to be conceived methodological developments that will further increase the efficiency of MT-DREAM$_{(ZS)}$.

2. Theory and Parallel Implementation

[13] Our method merges the strengths of differential evolution, sampling from past states, snooker updating,

randomized subspace exploration, and multiple-try Metropolis sampling [*Liu et al.*, 2000] for efficient high-dimensional posterior exploration. The resulting new code, MT-DREAM$_{(ZS)}$, is an extension of DREAM$_{(ZS)}$ (Vrugt et al., manuscript in preparation, 2012), and is especially designed for parallel implementation on a distributed computing cluster. We first describe multiple-try Metropolis sampling (MTM), and then continue with an detailed algorithmic description of MT-DREAM$_{(ZS)}$.

2.1. Multiple-Try Metropolis in MCMC Sampling

[14] For reasons stated earlier, it is particularly important to have an appropriate selection of the proposal distribution, $q(\mathbf{x}_{t-1}, \cdot)$, used to generate candidate points in each individual chain. Local moves (small jumps) have a higher chance of being accepted but explore only a small region. Large jumps, on the contrary, cover a larger part of the search space, yet are typically rejected. *Liu et al.* [2000] have introduced a general approach that directly confronts this tradeoff by creating multiple different candidate points simultaneously involving both small and large jumps. This multiple-try Metropolis (MTM) approach has several desirable advantages, one of them that the mixing of Markov chains is significantly enhanced. In this work, we use the so-called MTM(II) variant, which was found to be the most robust for a range of different posterior exploration problems [*Liu et al.*, 2000]. This MTM(II) approach assumes a symmetric proposal distribution, $q(\cdot)$, and can be described as follows.

[15] 1. Draw k trials $\mathbf{z}_1, \ldots, \mathbf{z}_k$ from $q(\mathbf{x}_{t-1}, \cdot)$, where $\mathbf{x}_{t-1}$ of size $1 \times d$ denotes the current state of the chain.

[16] 2. Compute the posterior density, $\pi(\mathbf{z}_j)$, of each of the k proposal points, $j = 1, \ldots, k$.

[17] 3. Randomly select one candidate point, $\mathbf{z}_j$ of $\mathbf{z}_1, \ldots, \mathbf{z}_k$ with probability proportional to $\pi(\mathbf{z}_j)$.

[18] 4. Draw $\mathbf{x}_1^*, \ldots, \mathbf{x}_{k-1}^*$ reference points from $q(\mathbf{z}, \cdot)$ and set $\mathbf{x}_k^* = \mathbf{x}_{t-1}$.

[19] 5. Accept $\mathbf{z}$ with probability

$$\alpha(\mathbf{x}_{t-1}, \mathbf{z}) = \min\left\{1, \frac{\pi(\mathbf{z}_1) + \ldots + \pi(\mathbf{z}_k)}{\pi(\mathbf{x}_1^*) + \ldots + \pi(\mathbf{x}_k^*)}\right\} \tag{4}$$

[20] This sampling scheme satisfies detailed balance, and therefore results in a reversible Markov chain with $\pi(\mathbf{x})$ as its stationary distribution [*Liu et al.*, 2000]. Numerical studies in the same paper have shown that MTM(II) is considerably more efficient than a traditional MH sampler. This is particularly inspiring considering the large amount of wasted samples. For each transition in each of the chain, $2k - 1$ samples are created of which only 1 is selected, and compared against the density of the current state of the chain. The information contained in the other $2k - 2$ points is simply thrown away, and can therefore be considered wasted. In response to this, *Frenkel* [2004] has proposed a MCMC sampling strategy that recycles information from such rejected states. This further increases the efficiency of posterior sampling, but this approach has not found widespread implementation and use.

[21] The use of multiple proposals in equation (4), however, places a heavier demand on computational resources, particularly when each candidate point is evaluated sequentially. For example, lets assume we use $k = 5$. For each transition in the Markov chain, this choice requires $5 + 4 = 9$ different evaluations of $\pi(\mathbf{x})$. Hence, the $k = 5$ proposal points need to be evaluated, together with $k - 1 = 4$ different points of the reference set. This is computationally rather demanding. For the same computational budget, a single-chain method is able to create 9 different transitions in the Markov chain. Indeed, our numerical tests do not corroborate the findings of *Liu et al.* [2000], but illustrate (not shown herein) that multiple-try RWM with a (multi)normal proposal distribution (MTRWMN) generally requires more function evaluations than standard RWMN to converge to the target distribution. This finding is consistent with the results of *Murray* [2007], who pointed out a critical deficiency of the work by *Liu et al.* [2000, p. 128, section 6.1] and also showed that if a similar proposal distribution is used, RWMN outperforms MTRWMN.

[22] In both these studies, the candidate points of the proposal and reference set have been evaluated sequentially. However, nothing prevents us from evaluating the k different proposal trials, followed by the $k - 1$ reference points simultaneously in parallel. This should significantly enhance the efficiency of MTM(II). For example, if these k different points are jointly evaluated then, in theory, parallel MTM(II) should be about $(2k - 1)/2$ more efficient than its sequential counterpart. Distributed computing thus significantly enhances the efficiency of posterior exploration, particularly when dealing with computationally demanding forward models [*Vrugt et al.*, 2006]. Yet, the efficiency of parallel MTM(II) remains essentially dependent on the choice of the proposal distribution used to generate transitions in each of the individual Markov chains. The MTM(II) method uses a rather simplistic and fixed (multivariate normal) proposal distribution to generate the points of the proposal and reference set. This is a potential source of inefficiency, in particular if the proposal distribution is a poor approximation of the target distribution [*Vrugt et al.*, 2003, 2008].

[23] We hypothesize that significant efficiency improvements can be made if the proposal distribution of MTM(II) is adaptively updated en route to the posterior target distribution. Such updating significantly enhances acceptance rate, and the speed at which the posterior distribution is explored. It therefore seems logical to merge the strengths of MTM(II) and DREAM and create a single MCMC sampler that combines automatic proposal updating with multi-try sampling and parallel computing to further enhance the efficiency of posterior sampling, and provide a general-purpose algorithm that can efficiently solve difficult and high-dimensional search and optimization problems. Unfortunately, standard DREAM requires at least $N = d/2$ to d chains to be run in parallel. This is a potential source of inefficiency, particularly for high-dimensional problems. For instance, for $k = 5$ and $d = 100$ parameters, we would need at least 250–500 different computational nodes to take full advantage of MTM(II), and accelerate the efficiency of posterior sampling. Most computer clusters will not readily have available such a large number of processors.

[24] To minimize computational requirements, we capitalize on recent developments in MCMC simulation and combine DREAM$_{(ZS)}$ with MTM(II). This new code, entitled multiple-try DREAM$_{(ZS)}$ or abbreviated MT-DREAM$_{(ZS)}$, uses DREAM$_{(ZS)}$ and MTM(II) as its main

building block, but creates multiple proposals simultaneously in each of the N chains by sampling from an archive of past states. Previous studies have shown that DREAM$_{(ZS)}$ achieves excellent sampling efficiencies for d up to 50–100 using only $N = 3$ different chains. Thus, if we create $k = 5$ different proposals in each individual chain, then MT-DREAM$_{(ZS)}$ would require only 15 different nodes for optimal performance. Indeed, this is a much lower number of nodes than would be required with MT-DREAM. The MT-DREAM$_{(ZS)}$ code is especially designed to solve complex, high-dimensional inverse problems and summarize model and parameter uncertainty. Section 2.2 provides a detailed algorithmic description of MT-DREAM$_{(ZS)}$, followed by four different case studies with increasing complexity.

2.2. Differential Evolution Adaptive Metropolis With Multiple-Try Sampling From an Archive of Past States

[25] We now describe our new code, entitled MT-DREAM$_{(ZS)}$, which uses MTM(II) and DREAM$_{(ZS)}$ as main building blocks.

[26] Let $\mathbf{Z} = [x_j^i]$ $(i = 1, \ldots, M_0; j = 1, \ldots, d)$ be a $M_0 \times d$ matrix, hereafter also referred to as archive, containing M_0 draws from the prior distribution, $p_d(\mathbf{x})$ of the d parameters. Similarly, let $\mathbf{X}$ be a $N \times d$ matrix defining the N initialized starting positions, $\mathbf{x}^i$, $i = 1, \ldots, N$ of the parallel chains by drawing samples from $p_d(\mathbf{x})$; $N \ll M_0$. Last, let T be the number of population evolution steps and k be the number of parallel proposals. The initial population $[\mathbf{X}_t; t = 0]$ is translated into a sample from the posterior distribution, $\pi(\mathbf{x})$ using the following pseudocode:

1. Set $M \leftarrow M_0$.

For $m \leftarrow 1, \ldots, T$ do (Population Evolution)

For $i \leftarrow 1, \ldots, N$ do (Chain Evolution: A. Proposal Step)

1a. Generate $l = 1, \ldots, k$ candidate points, $\mathbf{z}^{l,i}$ in chain i,

$$\mathbf{z}^{l,i} = \mathbf{x}^i + (\mathbf{1}_d + \mathbf{e}_d)\gamma(\delta, d')\left[\sum_{j=1}^{\delta}\mathbf{Z}^{r_1(j)} - \sum_{n=1}^{\delta}\mathbf{Z}^{r_2(n)}\right] + \boldsymbol{\epsilon}_d \quad (5)$$

where δ signifies the number of pairs used to generate the proposal, $\mathbf{Z}^{r_1}(j)$ and $\mathbf{Z}^{r_2(n)}$ are rows from the archive $\mathbf{Z}$; $r_1(j), r_2(n) \in \{1, \ldots, M\}$ and $r_1(j) \neq r_2(n)$. New values of $r_1(j)$ and $r_2(n)$ are sampled for each l. The values of $\mathbf{e}_d$ and $\boldsymbol{\epsilon}_d$ are drawn from $U_d(-b, b)$ and $N_d(0, b^*)$ with b and b^* small compared to the width of the target distribution, respectively, and the value of the jump size, γ, depends on δ and d', the number of dimensions that will be updated jointly (see next step).

1b. Replace each element $(j = 1, \ldots, d)$ of the $l = 1, \ldots, k$ parallel proposals $z_j^{l,i}$ with x_j^i using a binomial scheme with probability $1 - CR$,

$$z_j^{l,i} = \begin{cases} x_j^i & \text{if } U \leq 1 - CR,\ d' = d' - 1 \\ z_j^{l,i} & \text{otherwise} \end{cases} \qquad j = 1, \ldots, d \quad (6)$$

where CR denotes the crossover probability, and $U \in [0, 1]$ is a draw from a standard uniform distribution.

1c. Compute $\pi(\mathbf{z}^{l,i})$ for each of the $l = 1, \ldots, k$ proposals.

1d. Select $\mathbf{z}^i$ among the k proposals with probability $\pi(\mathbf{z}^i)$.

End for (Chain Evolution: A. Proposal Step)

For $i \leftarrow 1, \ldots, N$ do (Chain Evolution: B. Reference Step)

1e. Generate $l = 1, \ldots, k - 1$ reference points, $\mathbf{x}^{*,l,i}$ in chain i using equations (5) and (6) but now centered around $\mathbf{z}^i$,

$$\mathbf{x}^{*,l,i} = \mathbf{z}^i + (\mathbf{1}_d + \mathbf{e}_d)\gamma(\delta, d')\left[\sum_{j=1}^{\delta}\mathbf{Z}^{r_1(j)} - \sum_{n=1}^{\delta}\mathbf{Z}^{r_2(n)}\right] + \boldsymbol{\epsilon}_d \quad (7)$$

1f. Compute $\pi(\mathbf{x}^{*,l,i})$ for $l = 1, \ldots, k - 1$ and Set $\mathbf{x}^{*,k,i} = \mathbf{x}^i$ and $\pi(\mathbf{x}^{*,k,i}) = \pi(\mathbf{x}^i)$.

1g. Accept $\mathbf{z}^i$ with modified Metropolis acceptance probability:

$$\alpha(\mathbf{x}^{*,1,i}, \ldots, \mathbf{x}^{*,k,i}; \mathbf{z}^{1,i}, \ldots, \mathbf{z}^{k,i}) = \min\left\{1, \frac{\pi(\mathbf{z}^{1,i}) + \ldots + \pi(\mathbf{z}^{k,i})}{\pi(\mathbf{x}^{*,1,i}) + \ldots + \pi(\mathbf{x}^{*,k,i})}\right\} \quad (8)$$

1h. If accepted, move the chain to the candidate point, $\mathbf{x}^i = \mathbf{z}^i$, otherwise remain at the old location, $\mathbf{x}^i$.

End for (Chain Evolution: B. Reference Step)

End for (Population Evolution)

2. Append $\mathbf{X}$ To $\mathbf{Z}$ after each K steps and then update $M \leftarrow M + N$.

3. Compute the *Gelman and Rubin* [1992] convergence diagnostic, $\hat{R}_j$, for each dimension $j = 1, \ldots, d$ using the last 50% of the samples in each chain.

4. If $\hat{R}_j \leq 1.2$ for all j, stop and go to step 5; otherwise, go to population evolution.

5. Summarize the posterior pdf using $\mathbf{Z}$ after discarding the initial and burn-in samples.

[27] The MT-DREAM$_{(ZS)}$ algorithm is similar to DREAM, but uses multiple-try Metropolis sampling from an archive of past states to generate candidate points in each individual chain. This novel MCMC code has four main advantages. First, sampling from the past circumvents the requirement of using $N = d$ for posterior exploration. Especially for high-dimensional problems with large d, this has been shown to speed up convergence to a limiting distribution [*ter Braak and Vrugt*, 2008; Vrugt et al., manuscript in preparation, 2012]. Second, the parallel multiproposal implementation (see Figure 1) increases the efficiency of posterior exploration and accelerates the speed of convergence. Third, unlike DREAM, outlier chains do not require explicit consideration and removal. At any time during the simulation, transition from aberrant trajectories to the modal region remain possible by sampling their own immediate past state from $\mathbf{Z}$ in combination with $\gamma = 1$. The chance of such jumps increases with increasing length of $\mathbf{Z}$. This is highly desirable. Even during burn-in, the N trajectories simulated with MT-DREAM$_{(ZS)}$ therefore maintain detailed balance at every single step in the chain (see *ter Braak and Vrugt* [2008] and Vrugt et al. (manuscript in preparation, 2012) for proofs of detailed balance and ergodicity with sampling from past states). Finally, as the proposal jumps in DREAM$_{(ZS)}$ are generated from an archive of past states, MT-DREAM$_{(ZS)}$ does not require sequential updating of the individual chains $i = 1, \ldots, N$ as implemented in DE-MC

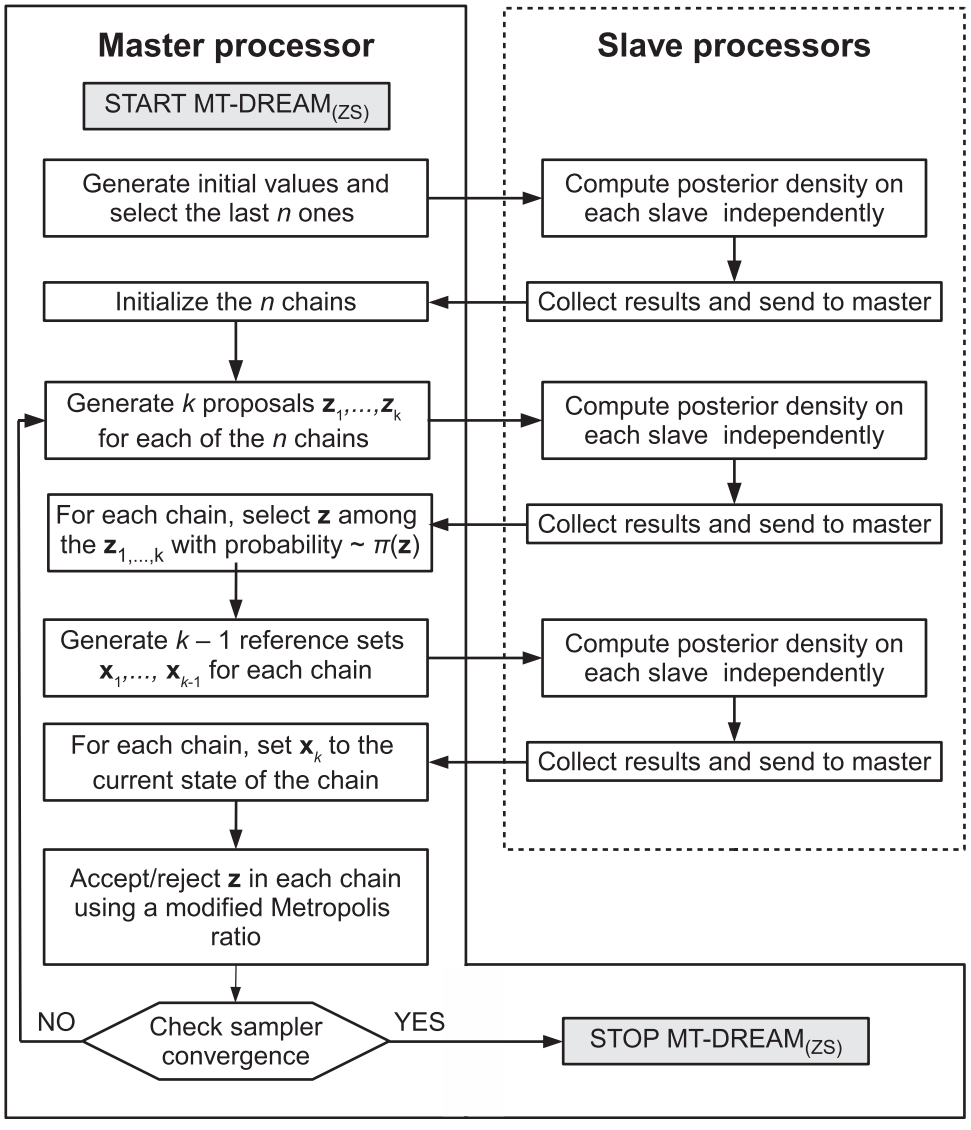

Figure 1. Sketch of the distributed MT-DREAM$_{(ZS)}$ algorithm.

and DREAM to ensure detailed balance. This is of great advantage in a multiprocessor environment because each proposal point can then be simultaneously evaluated on a different node. The official proof of reversibility of DE-MC and DREAM demands the chains to be updated sequentially which impairs parallelization. Finally, MT-DREAM$_{(ZS)}$ contains a snooker update to increase the diversity of the candidate points, details of which are given by *ter Braak and Vrugt* [2008], and Vrugt et al. (manuscript in preparation, 2012).

[28] To speed up convergence to the target distribution, MT-DREAM$_{(ZS)}$ estimates a probability density function of different *CR* values during burn-in so that the average jumping distance is maximized. In practice, the probability p_m of n_{CR} different crossover values, $CR = m/n_{CR} \,|\, m = 1, \ldots, n_{CR}$, is estimated by maximizing the squared distance, $\Delta = \sum_{i=1}^{N}\sum_{j=1}^{d}(\overline{\mathbf{x}}_{j,t}^{i} - \overline{\mathbf{x}}_{j,t-1}^{i})^2$ between the two subsequent samples $\overline{\mathbf{x}}_t$ and $\overline{\mathbf{x}}_{t-1}$ of the N different chains. The position of the chains is normalized with the standard deviation of each individual dimension (parameter) calculated from the current population, $\mathbf{X}_t$, so that all d dimensions contribute equally to Δ. The algorithm results in an optimized probability for each individual *CR* value. This distribution is determined during burn-in and used to randomly select a *CR* value for each different proposal point, which in turn determine the effective number of dimensions or d' used to calculate $\gamma(\cdot)$ in equations (5) and (7). A detailed description of this adaptation strategy is given by *Vrugt et al.* [2009] and so will not be repeated herein.

Theorem *Suppose* $\pi(\cdot)^N$ *is a fixed target probability distribution, on a state space* $\boldsymbol{\chi}$. *MT-DREAM*$_{(ZS)}$ *constructs a Markov chain kernel* $P(\cdot)$, *which has* $\pi(\cdot)^N$ *as its stationary distribution if* $T \rightarrow \infty$ *such that*

$$\|P_T(\mathbf{x}, \cdot) - \pi(\cdot)^N\| \rightarrow 0 \tag{9}$$

for any $\mathbf{x} \in \boldsymbol{\chi}$.

Proof We are left with giving a formal proof of detailed balance of MT-DREAM$_{(ZS)}$. In few words, DREAM$_{(ZS)}$

yields a Markov chain that is ergodic with unique stationary distribution with pdf $\pi(\cdot)^N$ because of its diminishing adaptation [*Roberts and Rosenthal*, 2007]. Indeed, the matrix $\mathbf{Z}$ grows during the course of the sampling process by an order $N/M = K/t$, which decreases in generation time t. The changes in the proposal distribution (and thus in the transition kernel $P(\cdot)$) therefore diminish with increasing length of $\mathbf{Z}$. A proof of reversibility of MTM(II) is given by *Liu et al.* [2000] and will be further detailed in Appendix A. Because both DREAM$_{(ZS)}$ and MTM(II) satisfy detailed balance, so thus their combination in MT-DREAM$_{(ZS)}$. This concludes our proof.

2.3. Selection of Algorithmic Variables in the MT-DREAM$_{(ZS)}$ Algorithm

[29] The MT-DREAM$_{(ZS)}$ algorithm contains several algorithmic variables that need to be specified by the user before the method can be used for posterior inference. These variables include N, the number of chains, k the number of parallel proposal trials in each individual chain, M_0, the initial size of the archive $\mathbf{Z}$, the thinning rate K used to periodically record samples in $\mathbf{Z}$, and p_{SK}, the probability of performing a snooker update (details are given by *ter Braak and Vrugt* [2008]). On the basis of recommendations by Vrugt et al. (manuscript in preparation, 2012) we set $N = 3 - 5$, $M_0 = 10d$ and $p_{SK} = 0.1$, and use default values of $\delta = 1$, $b = 0.05$, $b^* = 10^{-6}$, and $n_{CR} = 3$ [*Vrugt et al.*, 2008, 2009]. We vary the thinning rate, K, between the different case studies considered herein to make sure that sufficient draws are stored for posterior inference. From the guidelines of s_d in RWM, the optimal choice of $\gamma(\delta, d') = 2.38/\sqrt{2\delta d'}$ in equations (5) and (7). To help facilitate direct jumps between different disconnected posterior modes, we temporarily switch to $\gamma = 1$ at every 5^{th} proposal point [*Vrugt et al.*, 2008, 2009].

[30] This leave us with choosing a value for k. A few preliminary tests for a range of different search problems suggests that $k = 5$ works well in practice. We therefore recommend this value in future applications.

3. Cases Studies

[31] To illustrate the efficiency of MT-DREAM$_{(ZS)}$, we conducted a wide range of numerical experiments. These tests include two known mathematical target distributions, and two real-world studies involving the calibration of the Sacramento Soil Moisture Accounting (SAC-SMA) model [*Burnash*, 1995], and a 241-parameter groundwater model. These case studies cover a diverse set of problem features, including high dimensionality, nonlinearity, nonconvexity, multimodality, and numerous local optima. In all our calculations with MT-DREAM$_{(ZS)}$, we use the default settings of the algorithmic variables specified previously. We use $N = 3$ parallel chains, but temporarily switch to $N = 5$ for one of the case studies involving significant multimodality. To benchmark the results of MT-DREAM$_{(ZS)}$, we include comparison against the DREAM [*Vrugt et al.*, 2008, 2009], DREAM$_{(ZS)}$ (Vrugt et al., manuscript in preparation, 2012), and RWMN algorithms using standard settings of the algorithmic variables reported in the literature. Note that the RWMN sampler runs only a single chain, and uses a multivariate normal proposal distribution, $N_d(0, c\mathbf{I}_d)$ with c tuned to get an acceptance rate of about 24%. This is typically considered optimal [*Roberts et al.*, 1997].

[32] The algorithmic developments presented in this paper have been inspired by the increasing availability of distributed computing resources. Indeed, the potential of parallel computing sheds a completely different light on what constitutes an efficient algorithm. Widely celebrated search and optimization algorithms that have received a lot of attention in the past decades, might no longer be most efficient in a multiprocessor environment. Their inherent sequential topology limits multitasking. An example of this includes the single-chain AP [*Haario et al.*, 1999], AM [*Haario et al.*, 2001] and DRAM methods [*Haario et al.*, 2006]. Their current sequential topology prevents effective use of distributed computing resources. Multichain methods on the contrary are easier to parallelize but it is important to preserve detailed balance. This requirement dictates that the N different chains in DREAM are updated sequentially, and we therefore run this algorithm on a single processor. The DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ algorithms, on the contrary are specifically designed for implementation on a distributed computing network. Sampling from the past ensures reversibility even if the N chains and/or multiple candidate points are evaluated jointly in parallel. We therefore execute DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ in parallel using multiple different processors.

[33] The differences in computer implementation of the DREAM, DREAM$_{(ZS)}$, and MT-DREAM$_{(ZS)}$ algorithms complicates a comparative efficiency analysis. For instance, for the computational costs of a single proposal evaluation in DREAM, the DREAM$_{(ZS)}$ algorithm is able to execute N different candidate points simultaneously in parallel. Widely used measures such as the total number of function evaluations, hereafter referred to as *FE* is thus no longer sufficient to compare the efficiency of the different MCMC algorithms used herein. We therefore introduce a computational time unit (*CTU*). Sequential MCMC samplers, such as RWMN and DREAM, use one *CTU* for each *FE*, thus essentially $CTU = FE$. Parallel samplers, on the contrary, evaluate multiple proposal points in parallel, thus automatically $CTU < FE$. In particular, for DREAM$_{(ZS)}$, it is not difficult to demonstrate that $CTU = FE/N$. This leaves us with MT-DREAM$_{(ZS)}$. Unfortunately, it is not immediately obvious what the mathematical relationship is between *CTU* and *FE* for this particular algorithm. The proposal and reference steps (equations (5) and (7)) both need a single *CTU* but involve a different number of function evaluations. The proposal step simultaneously evaluates $N \times k$ points (i.e., *FE*), whereas the reference step executes $N \times (k-1)$ candidate points in parallel. A single *CTU* in MT-DREAM$_{(ZS)}$ is thus equivalent to approximately $N \times \left(k - \frac{1}{2}\right)$ *FE*, which after rearrangement gives $CTU = FE/\left(N \times \left(k - \frac{1}{2}\right)\right)$. Also, because the proposal and reference step in MT-DREAM$_{(ZS)}$ require 2 *CTUs*, MT-DREAM$_{(ZS)}$ is theoretically about 50% less efficient than DREAM$_{(ZS)}$. In practice, however, the multitry step will exhibit some important advantages, as will be demonstrated later. To make the comparison between DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ as fair as possible, our numerical experiments presented herein also include DREAM$_{(ZS)}$ with a number of chains similar to the number of parallel processors using by MT-DREAM$_{(ZS)}$, which is simply $N \times k$.

[34] Note that these developments essentially ignore the time required for communication between the master and slave nodes. In most practical applications involving CPU intensive forward models, the time it requires to communicate between the master and the slave nodes is negligibly small compared to the time it requires to execute the actual simulation model, and compute the desired output. In other words, the most efficient MCMC method requires the lowest number of *CTUs* to generate samples from the posterior distribution.

[35] We use three different diagnostic measures to check when convergence of each sampler to a limiting distribution has been achieved. The first diagnostic is the $\hat{R}$ statistic of *Gelman and Rubin* [1992] which compares the between and within variance of the different chains. Convergence is declared when $\hat{R}_j \leq 1.2$ for all $j = 1, \ldots, d$, and the corresponding *CTU* is denoted with $CTU_{\hat{R}}$.

[36] The second and third convergence diagnostic are derived using the approach of *Raftery and Lewis* [1992]. Suppose that we like to measure some posterior quantile, hereafter referred to as *qnt*. If we define a tolerance r of qnt and a probability s of being within that tolerance, the Raftery-Lewis diagnostic estimates the number of posterior samples, NT_{RL}, and the required burn-in length of the Markov chain, $BURN_{RL}$, necessary to satisfy the given tolerances. Yet the Raftery-Lewis diagnostic will differ depending on what quantile is being chosen. We therefore follow *El Adlouni et al.* [2006] and compute $BURN_{RL}$ and NT_{RL} for 9 different values of qnt $\in 0.1, 0.2, \ldots, 0.9$. We do this for each jth dimension ($j = 1, \ldots, d$) of the posterior target, and retain the largest values of $BURN_{RL}$ and NT_{RL}. We then report the number of *CTU* needed for the sampler to produce those required number of samples. We follow the statistical literature and set $r = 0.9$.

[37] The final performance criteria considered herein measures the autocorrelation between the various samples of the Markov chains created with the different MCMC algorithms. We use the so-called inefficiency factor (*IF*) [*Chib et al.*, 2002]. In principle, the *IF* is similar to the inverse of the numerical efficiency measure of *Geweke* [1992] and can be computed from the last samples in the Markov chains [*Chib et al.*, 2002]. We compute this *IF* criterion for each dimension and report the largest value. To be able to compare the values of *IF* for the different MCMC methods, we estimate this criterion using a similar number of (final) posterior samples. This number of samples is simply taken to be 25% of the maximum number of *FE* of the sequential codes.

[38] The four different performance criteria discussed so far primarily estimate the time required to reach convergence for each individual method, and generate high-quality (and thus uncorrelated) samples from the posterior distribution. These diagnostics essentially measure efficiency, without recourse to estimating the correctness of the sampled posterior distribution. For example, consider a case in which an algorithm has prematurely (quickly) converged to the wrong distribution. The convergence criteria would actually convey a spectacular performance. We therefore need to augment these three different efficiency diagnostics with criteria that explicitly measure the distance to the actual target distribution. Of course, such effectiveness criteria can only be computed if the posterior distribution is actually known beforehand. We consider two of such synthetic distributions in this paper, and introduce an additional diagnostic measure, D, that measures the average normalized Euclidean distance to the true mean μ_π and standard deviation σ_π of the posterior target distribution:

$$D = \sqrt{\frac{1}{2d}\sum_{i=1}^{d}\left[\left(\frac{\mu_\pi - \hat{\mu}_\pi}{\sigma_\pi}\right)^2 + \left(\frac{\sigma_\pi - \hat{\sigma}_\pi}{\sigma_\pi}\right)^2\right]} \quad (10)$$

where $\hat{}$ denotes the posterior moments derived from the posterior draws generated with each sampler. We take a similar number of (final) draws for each different MCMC method, and this number is identical to 25% of the total number of *FE* allowed for the sequential samplers. A similar calculation is used for *IF*. For our mathematical test distributions, we report the values of D alongside with $CTU_{\hat{R}}$, $BURN_{RL}$, NT_{RL}, and *IF*, In all our calculations reported herein, we do not consider the delayed rejection adaptive Metropolis algorithm (DRAM) [*Haario et al.*, 2006] because this adaptive sampler has shown to exhibit rather poor performance [*Vrugt et al.*, 2009].

3.1. A 200-Dimensional Multivariate Normal Distribution

[39] To test the performance of our code in the presence of high dimensionality, the first case study considers a 200-dimensional multivariate normal distribution, centered at the zero vector. The covariance matrix was set such that the variance of the jth variable was equal to j, with pairwise correlations of 0.5. The initial population is drawn from $\mathbf{X} \in [-5.0, 15.0]^d$ reflecting a lack of prior knowledge about the mean and variance of the posterior. A maximum total of 1,000,000 *CTU* were allowed for the sequential RWMN and DREAM methods. For the parallel DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ codes a maximum total of 400,000 *CTU* was deemed sufficient to explore the posterior target distribution. Thus, RWMN and DREAM were allowed to use more than two times the amount of time assigned to DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$.

[40] Table 1 presents summary statistics of 25 subsequent trials for each of the four different Metropolis samplers. The results presented in Table 1 highlight several important findings. First, MT-DREAM$_{(ZS)}$ provides the closest approximation of the actual target distribution. Although DREAM$_{(ZS)}$ with $N = 15$ chains converges the fastest of all the different algorithms, the resulting posterior samples not only exhibit considerably more autocorrelation, but also are less consistent with the true posterior distribution. Second, multiple-try sampling significantly enhances the mixing of the different Markov chains. Approximately 45% of the proposal points is being accepted with MT-DREAM$_{(ZS)}$, whereas an acceptance rate of about 17%–24% is found for the other MCMC samplers. This partly explains the superior performance of MT-DREAM$_{(ZS)}$. Note that acceptance rate of 45.2% obtained with MT-DREAM$_{(ZS)}$ falls within the range of 30%–50% recommended by *Liu et al.* [2000] for standard MTM. Third, and as anticipated, notice the inferior results of RWMN. The fixed proposal distribution (identity matrix), albeit scaled to receive an acceptance rate of 24%, is a rather poor approximation of the actual target distribution,

Table 1. Performance of MT-DREAM$_{(ZS)}$ Against RWMN, DREAM, and DREAM$_{(ZS)}$ for the 200-Dimensional Gaussian Distribution With Correlated Dimensions, Using a Maximum Total of 1,000,000 (RWMN and DREAM) and 400,000 (DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$) Computational Time Units (*CTU*)[a]

	N	D	$CTU_{\hat{R}}$ ($\times 10^4$ *CTU*)	$BURN_{RL}$ ($\times 10^3$ *CTU*)	NT_{RL} ($\times 10^6$ *CTU*)	*IF*	*AR* (%)
RWMN	1	0.524	N/A	30.29	10.541	807.5	24.3
DREAM	200	0.086	N/C	0.43	0.681	6.3	17.0
DREAM$_{(ZS)}$	3	0.062	3.975	0.21	0.447	121.0	16.8
DREAM$_{(ZS)}$	15	0.062	1.602	0.03	0.059	114.7	17.3
MT-DREAM$_{(ZS)}$	3	0.038	2.959	0.35	0.239	53.1	45.2

[a]N is the number of chains, D measures closeness to the true posterior target, $CTU_{\hat{R}}$, $BURN_{RL}$, and NT_{RL} are the *Gelman and Rubin* [1992] and the two *Raftery and Lewis* [1992] convergence criteria expressed in computational time, respectively, *IF* is the inefficiency factor, and *AR* is the average acceptance rate. D and *IF* are computed using the last 250,000 draws generated by each method within the allowed computational time. Reported values represent averages over 25 independent runs. N/A, not applicable; N/C, none of the 25 runs have converged within the allowed 10^6 *CTU*.

and hence many *FE* or *CTU* are necessary with RWMN to approximate the posterior distribution. Fourth, and as hypothesized in the introduction, the combination of parallel evaluation of the candidate points with sampling from the past in DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ tremendously reduces the required burn-in. Last, MT-DREAM$_{(ZS)}$ generates posterior samples with the smallest (auto)correlation among the different codes.

[41] To provide insights into the sampled posterior distributions, consider Figure 2, which presents histograms of dimension 1 and 200 of the multivariate normal distribution. The red lines represent the true marginal distributions of x_1 and x_{200}. The RWM samples receives particular poor performance, and the marginal distributions deviate considerably from their true counterpart. Multichain methods (second through fourth rows) receive a noticeable better performance, but overall MT-DREAM$_{(ZS)}$ receives superior results. The posterior distribution sampled with this method almost perfectly matches the actual target distribution.

3.2. A 25-Dimensional Trimodal Distribution

[42] The second case study involves a 25-dimensional trimodal pdf with three disconnected modes. This example builds on the bimodal distribution presented by *Vrugt et al.* [2009], and is given by $\pi(\mathbf{x}) = 3/6N_d(\mathbf{10}, \mathbf{I}_d) + 2/6N_d(\mathbf{5}, \mathbf{I}_d) + 1/6N_d(-\mathbf{5}, \mathbf{I}_d)$ where $\mathbf{10}$, $\mathbf{5}$, and $-\mathbf{5}$ are d-dimensional vectors. This distribution is notoriously difficult to approximate with MCMC simulation, because the three individual modes are so far separated that standard Metropolis samplers cannot jump from one mode to the other. This complicates convergence. A maximum total of 2,000,000 *CTU* was deemed sufficient for the sequential RWMN and DREAM algorithms to explore the target distribution. The DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ codes on the contrary were allowed to only use 400,000 *CTU*. This is still sufficient to reach convergence to a limiting distribution. Thus, the parallel codes consume only 1/5 of the total computational time that is assigned to RWMN and DREAM.

[43] Table 2 summarizes the performance of RWMN, DREAM, DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$. Listed statistics denotes averages over 25 independent trials. The results presented in Table 2 are qualitatively very similar to those of the previous study, and even more clearly highlight the excellent performance of MT-DREAM$_{(ZS)}$. As expected, RWMN exhibits a particular poor performance. Each different trial converges to a different posterior mode resulting in a rather poor approximation of the target distribution and thus rather high value of the D statistic. A single chain is typically unable to cope with this multimodal search space, and provide an accurate characterization of the target distribution. As expected, significantly better results are obtained if multiple different trajectories are run simultaneously. Not only, does the discrete proposal distribution of equation (3) allow for immediate jumps between the different disconnected modes, the use of a number of different chains also protects against premature convergence. Yet standard DREAM and DREAM$_{(ZS)}$ exhibit a rather poor acceptance rate. Less than 1 out of 10 candidate points is being accepted which causes a rather slow mixing of the individual Markov chains. Much better results are obtained with MT-DREAM$_{(ZS)}$ when multiple candidate points are jointly considered in each individual chain. This not only significantly increases the acceptance rate to about 30% but also results in the closest approximation of the target distribution.

[44] Figure 3 presents the results of our analysis. Figures 3a–3c present histograms of the sampled x_1 values using the DREAM (Figure 3a), DREAM$_{(ZS)}$ (Figure 3b), and MT-DREAM$_{(ZS)}$ (Figure 3c) algorithms. The red line depicts the true marginal posterior pdf of the trimodal test function. The results conclusively show that MT-DREAM$_{(ZS)}$ exhibits the best performance and provides the closest approximation of the true target distribution. This finding is consistent with the results reported in Table 2, and inspires confidence in the ability of MT-DREAM$_{(ZS)}$ to deal with multimodal posterior distributions.

[45] Figure 3d shows a trace plot of the sampled x_1 values derived with MT-DREAM$_{(ZS)}$. Each of the N Markov chains is coded with a different color. The different parallel trajectories mix well, and jump back and forth between the different posterior modes. The density of the points in each mode is consistent with the weight of each peak in the actual target distribution. These findings highlight the ability of MT-DREAM$_{(ZS)}$ to efficiently explore multimodal posterior distributions.

3.3. The Rainfall-Runoff Transformation: SAC-SMA Model

[46] Our first real-world case study considers calibration of the Sacramento Soil Moisture Accounting (SAC-SMA) model [*Burnash*, 1995]. The SAC-SMA model is a lumped conceptual watershed model that describes the transformation from rainfall into basin runoff using six different reservoirs (state variables). A unit hydrograph is commonly

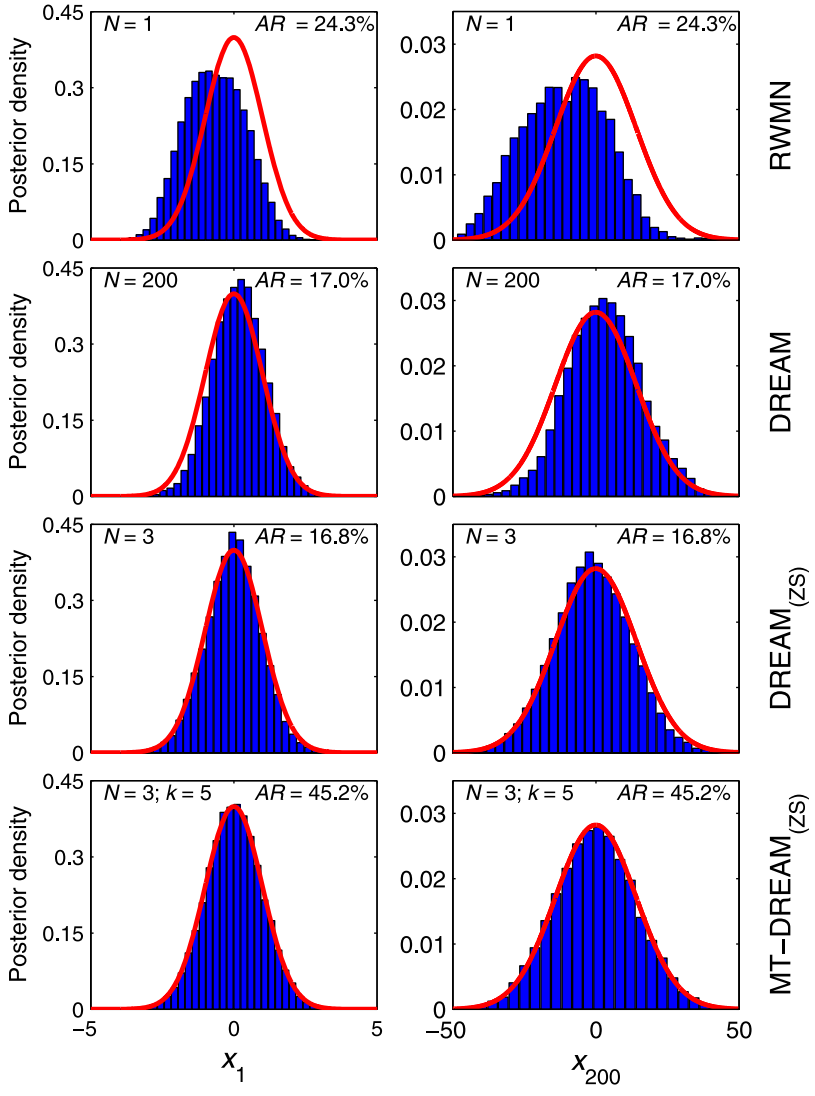

Figure 2. Marginal posterior probability density functions (pdfs) of dimensions (left) 1 (x_1) and (right) 200 (x_{200}) for the 200-dimensional multivariate normal distribution with correlated dimensions using RMWN (first row), DREAM (second row), DREAM$_{(ZS)}$ (third row), and MT-DREAM$_{(ZS)}$ (fourth row). Each individual plot reports N, the number of chains, and AR, the average acceptance rate. For MT-DREAM$_{(ZS)}$, we also list the value of k, the number of parallel proposal points. The true marginal distributions are indicated with a solid red line.

Table 2. Performance of MT-DREAM$_{(ZS)}$ Against RWMN, DREAM, and DREAM$_{(ZS)}$ for the 25-Dimensional Trimodal Distribution and a Maximum Total of 2,000,000 (RWMN and DREAM) and 400,000 (DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$) Computational Time Units (CTU)[a]

	N	D	$CTU_{\hat{R}}$ ($\times 10^4$ CTU)	$BURN_{RL}$ ($\times 10^3$ CTU)	NT_{RL} ($\times 10^6$ CTU)	IF	AR (%)
RWMN	1	0.830	N/A	0.24	0.503	8.1	24.1
DREAM	25	0.087	113.041	16.24	32.652	73.4	5.9
DREAM$_{(ZS)}$	5	0.191	3.640	7.25	12.841	120.9	9.8
DREAM$_{(ZS)}$	25	0.085	7.836	0.56	0.894	50.1	9.8
MT-DREAM$_{(ZS)}$	5	0.052	3.360	0.85	1.768	196.1	28.6

[a]N is the number of chains, D measures closeness to the true posterior target, $CTU_{\hat{R}}$, $BURN_{RL}$, and NT_{RL} are the *Gelman and Rubin* [1992] and the two *Raftery and Lewis* [1992] convergence criteria expressed in computational time, respectively, IF is the inefficiency factor, and AR is the average acceptance rate. D and IF are computed using the last 500,000 draws generated by each method within the allowed computational time. Reported values represent averages over 25 independent runs. N/A, not applicable.

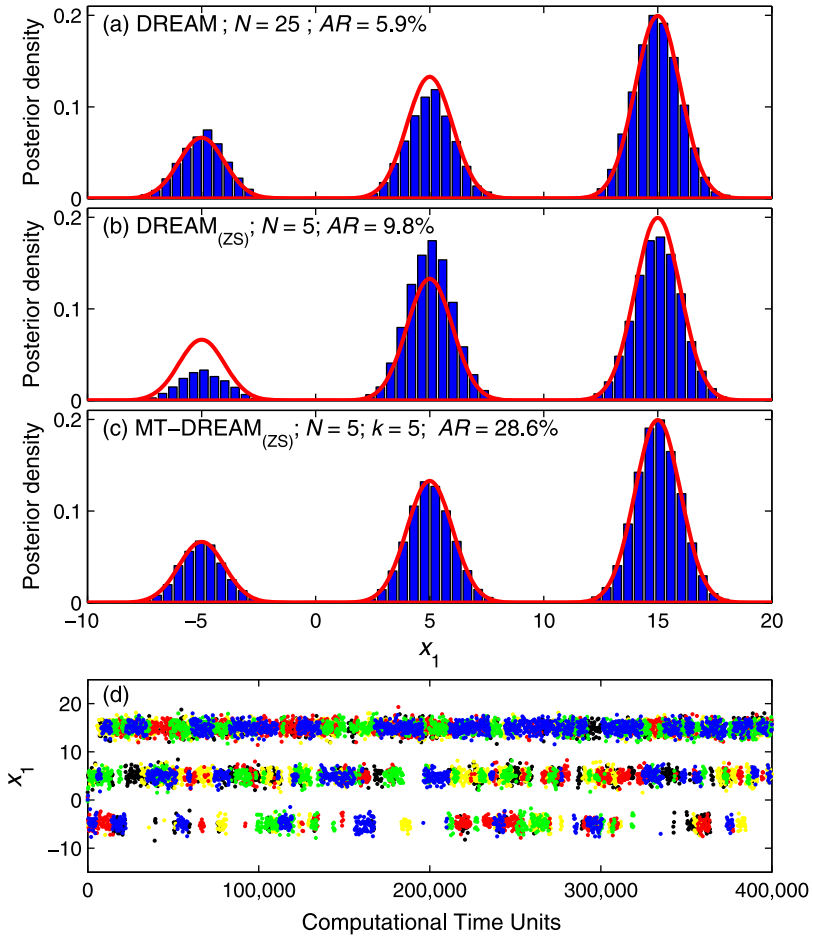

Figure 3. Marginal posterior pdfs of dimension 1 (x_1) obtained with (a) DREAM, (b) DREAM$_{(ZS)}$, and (c) MT-DREAM$_{(ZS)}$. The variable N is the number of chains, and AR denotes the average acceptance rate. For MT-DREAM$_{(ZS)}$, the value of k, the number of parallel proposal points, is also listed. The histograms are derived from the last 50% of the samples in the joint chains. The true posterior pdf is indicated with the solid red line. (d) The evolution of the N pathways sampled with MT-DREAM$_{(ZS)}$. These results demonstrate the superior performance of MT-DREAM$_{(ZS)}$.

used to rout channel inflow downstream and compute streamflow at the gauging point. This model is extensively used by the National Weather Service for flood forecasting throughout the United States, and has 13 user-specifiable (and 3 fixed) model parameters, which are listed in Table 3. Inputs to the model include mean areal precipitation (MAP) and potential evapotranspiration (PET) while the outputs are estimated evapotranspiration and channel inflow. Various studies have demonstrated that calibration of the SAC-SMA model is very difficult because of the presence of numerous local optima in the parameter space with both small and large domains of attraction, discontinuous first derivatives, and curving multidimensional ridges [*Duan et al.*, 1992; *Thiemann et al.*, 2001; *Vrugt et al.*, 2006, 2009; *Chu et al.*, 2010]. Although this study only involves 13 different parameters, it poses an interesting challenge for MCMC samplers, as will be shown later in section 3.3.

[47] We estimate the posterior distribution of the SAC-SMA parameters using historical data from the Leaf River watershed. This humid basin of approximately 1950 km^2 is located north of Collins, Mississippi, in the Unites States. We used 2 years of daily discharge data from 1 January 1953 to 31 December 1954 to estimate the SAC-SMA parameters. In practice, it is advisable to use a longer record of calibration data [*Yapo et al.*, 1996; *Vrugt et al.*, 2006], but deliberately we use only a 2 year record to result in a very

Table 3. Description of the SAC-SMA Model Parameters, Including Their Prior and 95% Posterior Uncertainty Intervals Derived With MT-DREAM$_{(ZS)}$

Parameter	Units	Prior	Posterior
UZTWM	mm	1.0–150.0	14.8–37.6
UZFWM	mm	1.0–150.0	15.2–33.9
LZTWM	mm	1.0–500.0	222.7–274.7
LZFPM	mm	1.0–1000.0	82.7–134.1
LZFSM	mm	1.0–1000.0	30.7–89.1
ADIMP		0.0–0.40	0.23–0.37
UZK	day^{-1}	0.1–0.5	0.24–0.49
LZPK	day^{-1}	0.0001–0.025	0.012–0.025
LZSK	day^{-1}	0.01–0.25	0.22–0.25
ZPERC		1.0–250.0	144.1–249.0
REXP		1.0–5.0	2.45–4.90
PCTIM		0.0–0.1	6.9×10^{-5} to 0.001
PFREE		0.0–0.6	0.004–0.210

challenging parameter estimation problem with multiple disconnected regions of attraction [*Vrugt et al.*, 2009]. We assume a flat or uniform prior distribution of the SAC-SMA model parameters with ranges specified in Table 3.

[48] The following posterior density function, $p(\mathbf{x}|\hat{\mathbf{y}}, \sigma_e, \boldsymbol{\phi})$, was used to compare the SAC-SMA modeled streamflow dynamics with the observed discharge data [*Box and Tiao*, 1973; *Thiemann et al.*, 2001]:

$$p(\mathbf{x}|\hat{\mathbf{y}}, \sigma_e, \boldsymbol{\phi}) = \prod_{i=1}^{N_m} \frac{w(\beta)}{\sigma_e} \exp\left[-c(\beta)\left|\frac{y_i(\mathbf{x}, \boldsymbol{\phi}) - \hat{y}_i}{\sigma_e}\right|^{\frac{2}{1+\beta}}\right] \quad (11a)$$

$$c(\beta) = \left\{\frac{\Gamma[3(1+\beta)/2]}{\Gamma[(1+\beta)/2]}\right\}^{\frac{2}{1+\beta}} \quad (11b)$$

$$w(\beta) = \frac{\{\Gamma[3(1+\beta)/2]\}^{\frac{1}{2}}}{(1+\beta)\{\Gamma[(1+\beta)/2]\}^{\frac{3}{2}}} \quad (11c)$$

where $y_i(\mathbf{x}, \boldsymbol{\phi})$ ($\hat{y}_i$) denotes the SAC-SMA simulated (observed) streamflow, N_m signifies the total number of measurements, $\boldsymbol{\phi}$ represents the initial and forcing conditions, $\beta \in [0, 1]$ defines the kurtosis of the density, and σ_e is the measurement error. We assume a normal density, $\beta = 0$, and estimate σ_e jointly with $\mathbf{x}$, the model parameters. This is a common approach in statistics, if detailed information about the measurement error is lacking. A maximum total of 100,000 *CTU* was used to approximate the posterior distribution of σ_e and the 13 SAC-SMA model parameters.

[49] Our previous work [*Vrugt et al.*, 2009] has demonstrated the superiority of DREAM over other adaptive MCMC samplers including the optimal RWMN sampler, DRAM [*Haario et al.*, 2006] and DE-MC [*ter Braak*, 2006]. This work also illustrated a rather poor performance of the Shuffled Complex Evolution (SCE-UA) global optimization algorithm of *Duan et al.* [1992]. SCE-UA was originally developed in the early 1990s to solve highly nonlinear, nonconvex and noncontinuous optimization problems, and because of its efficiency and effectiveness has become the method of choice for watershed model calibration problems. Because the actual posterior distribution for this real-world problem is not known beforehand, the "best" solution found with SCE-UA was used as benchmark to test the performance of the different MCMC schemes. Indeed, the target distribution should center around the SCE-UA solution. Yet, for this particular problem, SCE-UA was found to get stuck in a local basin of attraction with RMSE values of about 13.7 $m^3\ s^{-1}$, whereas DREAM identified a global minima around 13.2 $m^3\ s^{-1}$. This difference in RMSE appears rather marginal, but the associated SAC-SMA parameter values derived with SCE-UA are substantially removed from their posterior distribution derived with DREAM. Repeated trials with SCE-UA using different values of the algorithmic variables yielded very similar results, and did not resolve the problem with premature convergence. We therefore exclude SCE-UA from our analysis, and instead consider the SP-UCI method of *Chu et al.* [2010]. This new global optimizer was especially designed to overcome some of the main flaws of SCE-UA. The results of *Chu et al.* [2010], although limited to a few case studies, demonstrate the advantages of SP-UCI over the standard SCE-UA method. *Chu et al.* [2010] kindly shared the SP-UCI code with us, and we ran this optimizer sequentially using standard settings of the algorithmic variables and the default sum of squared error (SSR) objective function. Like RWMN and DREAM, each *FE* with SP-UCI thus consumes a single *CTU*.

[50] Table 4 summarizes the results of the different algorithms used herein. The reported statistics denote averages over 25 different trials. We draw two main conclusions from the tabulated statistics. In the first place, notice that the three different MCMC methods exhibit a very similar effectiveness. The DREAM, $\text{DREAM}_{(ZS)}$MCMC samplers, as will be shown later on and $\text{MT-DREAM}_{(ZS)}$ algorithms consistently converge to the approximate same invariant distribution, with a negligibly small variability among the 25 different trials. A second interesting finding, perhaps rather unexpected, is that SP-UCI performs rather poorly with RMSE values that are substantially larger than those reported for the different MCMC algorithms. About 95% of the SP-UCI trials converge prematurely and get stuck in a local basin of attraction en route to the global minimum (maximum likelihood) of the posterior distribution. This reinforces the severity of the SAC-SMA model calibration problem, and questions the ability of SP-UCI to deal with complex and multimodal search spaces.

[51] A third, and final observation is that $\text{MT-DREAM}_{(ZS)}$ is most efficient in generating posterior samples. The $CTU_{\hat{R}}$, $BURN_{RL}$, and NT_{RL} convergence diagnostics demonstrate that $\text{MT-DREAM}_{(ZS)}$ requires the least amount of computational time to explore the posterior

Table 4. Comparison of $\text{MT-DREAM}_{(ZS)}$ Against DREAM, $\text{DREAM}_{(ZS)}$, and SP-UCI for the SAC-SMA Model Calibration and a Maximum Total of 100,000 Computational Time Units (*CTU*)[a]

	N	E_{MEAN} ($m^3\ s^{-1}$)	E_{MIN} ($m^3\ s^{-1}$)	E_{MAX} ($m^3\ s^{-1}$)	N_{LOC}	$CTU_{\hat{R}}$ ($\times 10^4$ *CTU*)	$BURN_{RL}$ ($\times 10^3$ *CTU*)	NT_{RL} ($\times 10^6$ *CTU*)	*IF*	*AR* (%)
DREAM	13	13.1	13.1	13.2	0	8.722[b]	118.85	119.233	261.2	5.2
$\text{DREAM}_{(ZS)}$	3	13.1	13.1	13.2	0	0.846	22.39	21.321	743.5	6.7
$\text{DREAM}_{(ZS)}$	15	13.1	13.1	13.2	0	1.946	49.24	79.449	1350.0	5.1
$\text{MT-DREAM}_{(ZS)}$	3	13.1	13.1	13.2	0	0.566	9.64	14.783	478.8	24.0
SP-UCI	N/A	13.6	13.1	14.0	24	N/A	N/A	N/A	N/A	N/A

[a]Listed statistics represent averages from 25 different trials. The variables N, $CTU_{\hat{R}}$, $BURN_{RL}$, NT_{RL}, *IF*, and *AR* have been defined in the main text and previous tables. *IF* is computed using the last 25,000 draws generated by each method within the allowed computational time. E_{MEAN}, E_{MIN}, and E_{MAX} denote the average, minimum, and maximum best root-mean-square errors (RMSE) of the 25 trials. The MCMC *SE* being in the range 0.04–0.07 $m^3\ s^{-1}$ for the Metropolis samplers, RMSE values are rounded to the first decimal. The variable N_{LOC} reports the number of runs being stuck into a local minima (RMSE > 13.3 $m^3\ s^{-1}$). N/A, not applicable.

[b]Of the 25 runs, 14 did not appropriately converge within the allowed 10^5 *CTU*.

distribution of the SAC-SMA model parameters. According to the $CTU_{\hat{R}}$ criterion, MT-DREAM$_{(ZS)}$ is about twice as efficient as the most efficient DREAM$_{(ZS)}$ parameterization ($N = 3$), and considerably more efficient than the original DREAM algorithm. This again highlights the advantages of multiple-try sampling. The acceptance rate of MT-DREAM$_{(ZS)}$ of about 25% is about 4–5 times higher than the other adaptive Metropolis samplers. This speeds up the efficiency of posterior exploration, and enables MT-DREAM$_{(ZS)}$ to cope with difficult response surfaces.

[52] Although the different diagnostic metrics convey useful information about the performance of the different algorithms, it remains useful to study the convergence behavior of the different methods in more detail. Consider Figure 4, which depicts the evolution of the sampled values of the upper zone tension water maximum storage (UZTWM) parameter (see, e.g., *Thiemann et al.* [2001] for details) using the DREAM (Figure 4a), DREAM$_{(ZS)}$ (Figure 4b), MT-DREAM$_{(ZS)}$ (Figure 4c), and SP-UCI (Figure 4d) optimization algorithms. Indeed, the results presented in the various panels are consistent with our previous conclusions. The MT-DREAM$_{(ZS)}$ code converges most rapidly and requires about 4,000 *CTU* to explore the posterior target distribution. This is significantly more efficient than the other two MCMC codes which require about 8,000 (DREAM$_{(ZS)}$) and 65,000 (DREAM) *CTU* to converge to a limiting distribution. Finally, SP-UCI converges rather quickly, but only about 5% of the trials finds the appropriate values of UZTWM.

3.4. Groundwater Model Calibration at the Nevada Test Site: 241 Parameters

[53] The final case study presented herein revisits the work of *Keating et al.* [2010], and involves calibration and predictive uncertainty analysis of a highly parameterized and strongly nonlinear groundwater model. This requires specification of $d = 241$ different parameters, and constitutes a very difficult optimization problem. A detailed description of the model, site, and data are given by

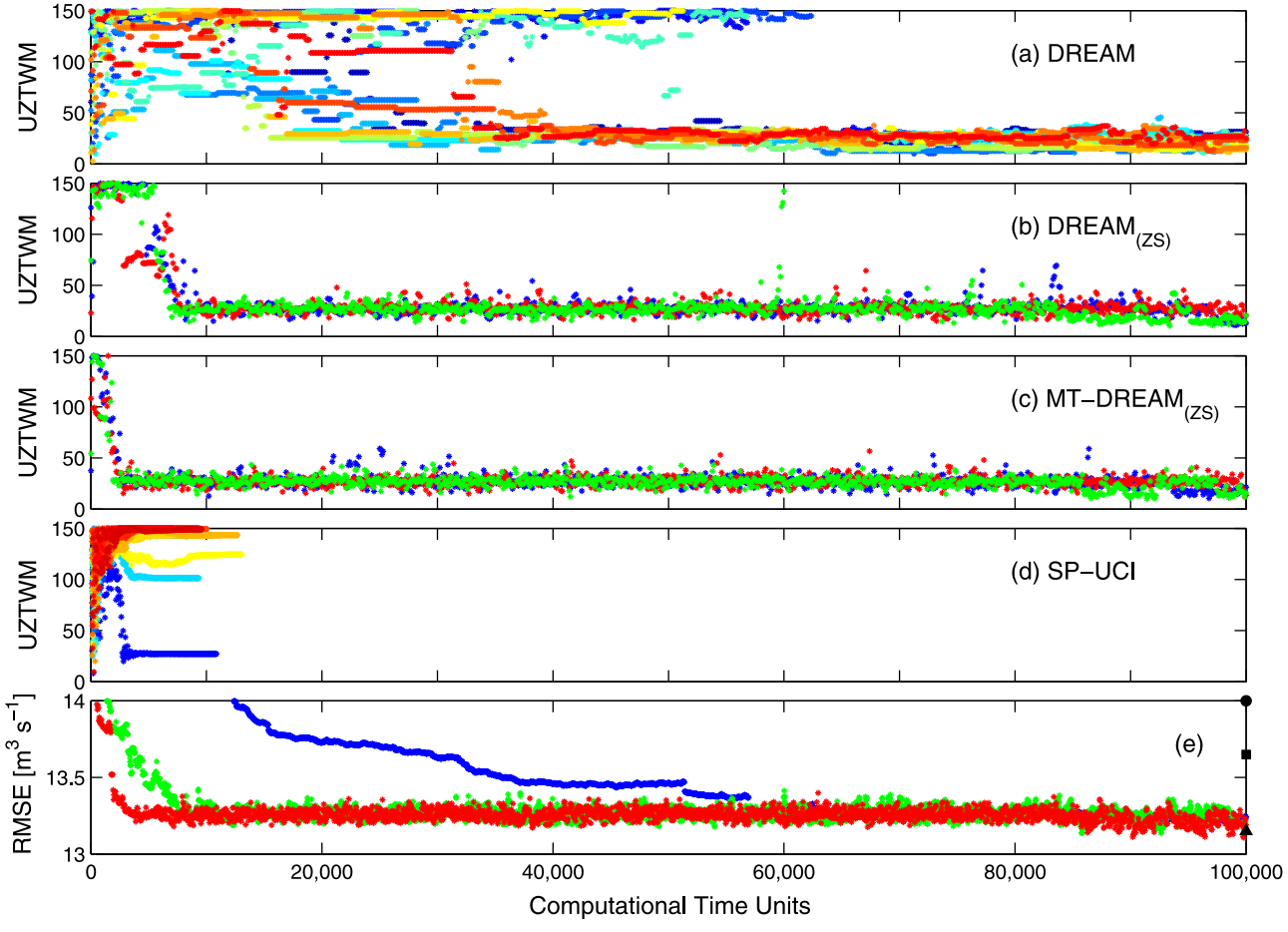

Figure 4. Evolution of sampled values of the upper zone tension water maximum storage (UZTWM) parameter (in mm) with the (a) DREAM, (b) DREAM$_{(ZS)}$ with $N = 3$, and (c) MT-DREAM$_{(ZS)}$ MCMC sampling schemes and (d) SP-UCI global optimization algorithm. Each chain in Figures 4a–4c is coded with a different color. For SP-UCI, we use color coding to illustrate each of the 25 different runs. (e) The evolution of the mean root-mean-square error (RMSE) value of the N different chains derived with DREAM (blue), DREAM$_{(ZS)}$ with $N = 3$ (green), and MT-DREAM$_{(ZS)}$ (red). The black square, black dot, and black triangle in Figure 4e represent the mean, maximum, and minimum RMSE value of the 25 different SP-UCI runs, respectively.

Keating et al. [2010] and so will not be repeated herein. We merely provide a brief synopsis.

[54] Between 1951 and 1992, a total of 659 underground nuclear tests were conducted in Yucca Flat, Nevada Test Site, United States. These explosions have likely enhanced the flux of water into the lower aquifer. A complex three-dimensional hydrogeological flow and transport model was setup to analyze whether the underground tests potentially increased the flux of radionuclides to the groundwater. Yet, this detailed model was shown to be too CPU intensive to benefit from state-of-the-art optimization and uncertainty analysis methods. A fast-running surrogate model was therefore developed that mimics the key characteristics of the full process model, but is computationally way more efficient.

[55] The design of this surrogate model was based on the simple assumption that for a specific head increase at the location of an explosion, the immediate head perturbation imposed at a nearby well will be determined by only (1) the distance of the well from the test, (2) the time elapsed since the test, and (3) the properties of the rock at the point of measurement. This simplified model contains 241 different parameters, of which 221 are nuclear testing effect parameters, 10 are rock permeabilities and another 10 are associated with groundwater recharge. As calibration data set we used 361 different head measurements collected at 60 different wells and spanning the time period from 1958 to 2005. In keeping with the previous study, a weighted sum of square residuals (WSSR) was used to measure the distance between the measured head data, $\hat{y}_i$ and respective predictions, $y_i(\mathbf{x}, \phi)$ of the model:

$$\mathrm{WSSR}(\mathbf{x}|\hat{\mathbf{y}}, \boldsymbol{\phi}) = \sum_{i=1}^{N_m} [u_i \left(y_i(\mathbf{x}, \boldsymbol{\phi}) - \hat{y}_i \right)]^2 \quad (12)$$

where u_i are weighting factors (for assignment of weights, see *Keating et al.* [2010]).

[56] Previous results presented by *Keating et al.* [2010] have shown that DREAM required about 50 million surrogate model evaluations to converge to a limiting parameter distribution with WSSR values ranging between 370 and 430. This distribution, however cannot be considered the actual posterior distribution. In the same paper, it was shown that null-space Monte Carlo (NSMC) [*Tonkin and Doherty*, 2009] converged to a very similar distribution of parameter values, but found one realization with a significantly better WSSR value of about 186 after 402,586 surrogate model evaluations. Finding this solution was by no means easy. It required a joint use of manual and computer-based parameter estimation methods, including the covariance matrix adaptation evolutionary strategy (CMAES) [*Hansen et al.* 2003], truncated singular value decomposition (SVD) [*Tonkin and Doherty*, 2005; *Marquardt*, 1963], and automatic user intervention (AUI) [*Doherty*, 2009], all being part of the PEST optimization toolbox [*Doherty*, 2009]. We now evaluate the performance of SP-UCI, DREAM$_{(ZS)}$, and MT-DREAM$_{(ZS)}$, and test whether they are able to locate the minimum WSSR value of about 186. We believe that comparison of DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ against global optimization techniques such as PEST and SP-UCI is necessary to benchmark and test whether the MCMC schemes considered herein actually converge to the appropriate posterior distribution, and not some local basin of attraction.

[57] Each algorithm was run with a flat prior distribution. The SP-UCI algorithm works directly with the WSSR objective function, but DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ require a log-likelihood function. We modify equation (12) and use the following posterior density function:

$$p(\mathbf{x}|\hat{\mathbf{y}}, \boldsymbol{\phi}) \propto -\frac{1}{2} \ln \left[(2\pi)^{N_m} |\boldsymbol{\Sigma}| \right] - \frac{1}{2} \sum_{i=1}^{N_m} [u_i \left(y_i(\mathbf{x}, \boldsymbol{\phi}) - \hat{y}_i \right)]^2 \quad (13)$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix with nonnegative diagonal elements taken as $(u_i)^{-2}$. Note that equation (13) just provides a different scaling of the WSSR objective function, and thus by no means affects the properties of the response surface and location of the posterior parameter distribution.

[58] A maximum total of 1,750,000 *CTU* was allowed for the different calibration and posterior sampling methods using standard settings of the algorithmic variables.

[59] Figure 5 presents the evolution of the sampled WSSR values for SP-UCI (Figure 5a), DREAM$_{(ZS)}$ (Figure 5a), and MT-DREAM$_{(ZS)}$ (Figure 5c). Each color represents the path of a different trial (SP-UCI) or Markov chain (DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$). The triangle, square, and cross at the right-hand side of Figure 5c report the minimum WSSR values found with SP-UCI, DREAM, and PEST, respectively. The advantages of multiple-try sampling are immediately obvious. MT-DREAM$_{(ZS)}$ has converged to a limiting distribution after about 180,000 *CTU* with WSSR values around 200. The minimum WSSR value of 193 found with MT-DREAM$_{(ZS)}$ is very close to the value of 186 found by PEST. This is a remarkable result, and inspires confidence in the ability of MT-DREAM$_{(ZS)}$ to solve highly parameterized inversion problems. The performance of SP-UCI is rather poor. The code has stagnated after about 50,000 *CTU* to WSSR values of about 910, and is unable to escape from this local basin of attraction. Population degeneration has caused SP-UCI to prematurely converge. An attempt to introduce particle diversity after about 150,000 *CTU* (scatter), remains unsuccessful even after 400,000 *CTU* (not shown). Finally, DREAM$_{(ZS)}$ rapidly converges to WSSR values between 400 and 500, and cannot seem to get out of this subspace of solutions. About 1.5 million of additional *CTU* (not shown) are necessary for one chain to find WSSR values of around 200, but official convergence to the posterior target requires many additional *CTU* (not shown). This finding holds for DREAM$_{(ZS)}$ parameterized with both $N = 3$ and $N = 15$ parallel chains.

[60] The results in Figure 5 not only indicate superior performance of MT-DREAM$_{(ZS)}$ but also highlight the advantages of stochastic search, and sampling from the past. Whereas the population diversity of SP-UCI quickly deteriorates, the Metropolis selection rule of MT-DREAM$_{(ZS)}$ naturally maintains variability as it is especially designed to converge to a distribution of parameter values, rather than a single solution. The various chains simulated with MT-DREAM$_{(ZS)}$ therefore refuse to settle on a single point, and continue to explore a range of WSSR values. This ability to maintain adequate variability is a necessity to be able to solve complicated search and optimization problems. The variability sampled with DREAM$_{(ZS)}$

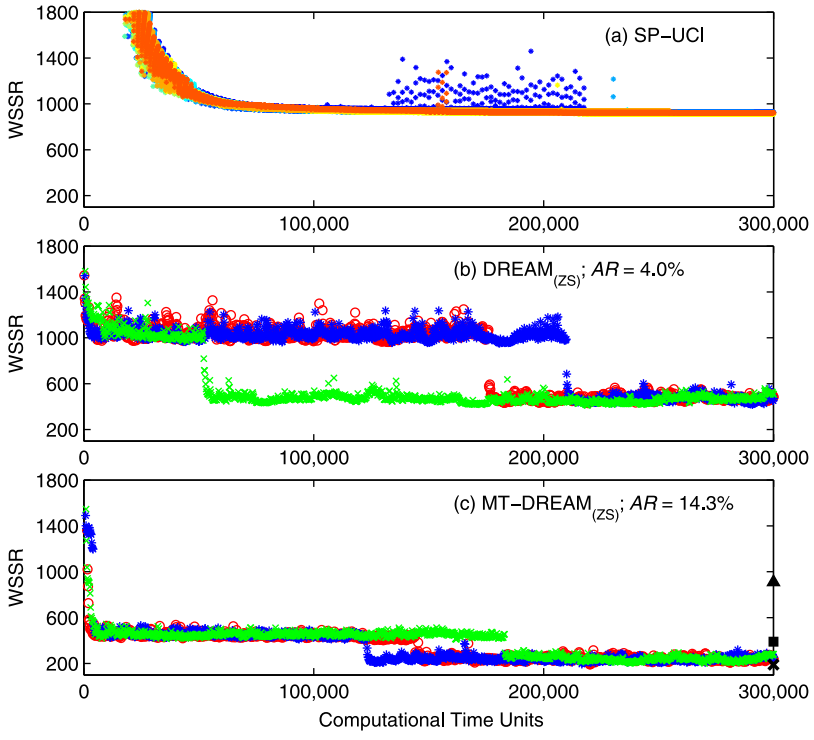

Figure 5. Trace plots of the sampled weighted sum of square residuals (WSSR) values using (a) SP-UCI, (b) DREAM$_{(ZS)}$, and (c) MT-DREAM$_{(ZS)}$ for the 241-dimensional groundwater model calibration problem. The variable *AR* has been defined in the text. Colors are used to indicate different trials (SP-UCI) or Markov chains (DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$). The different symbols at the right-hand side of Figure 5c summarize the minimum WSSR values found with SP-UCI (triangle), DREAM (square), and PEST (cross).

and MT-DREAM$_{(ZS)}$ is further enhanced by generating jumps from an archive of past solutions.

[61] This is further illustrated in Figure 6, which for MT-DREAM$_{(ZS)}$ presents the evolution of the $\hat{R}_j$ statistic of *Gelman and Rubin* [1992] for each individual parameter, $j = 1, \ldots, d$. As stated earlier, a value of $\hat{R}_j < 1.2$ is typically used to declare convergence to a limiting distribution. Whereas the sampled WSSR values stabilize after about 180,000 *CTU*, MT-DREAM$_{(ZS)}$ has not formally converged until approximately 1 million *CTU*. It simply takes such a large number of additional samples to adequately explore the $d = 241$ dimensional posterior distribution.

[62] The new results with MT-DREAM$_{(ZS)}$ likely alter some of the previous conclusions of *Keating et al.* [2010], who compared formal Bayesian inference with DREAM [*Vrugt et al.*, 2008] against the more efficient NSMC method of *Tonkin and Doherty* [2009] that does not maintain detailed balance and therefore has no theoretical Bayesian foundation. Our previous results demonstrated that DREAM and NSCM converged to very similar marginal (posterior) parameter distributions. Yet, our results presented herein demonstrate substantial differences between the marginal parameter distributions inferred with MT-DREAM$_{(ZS)}$ and their respective pdfs previously derived

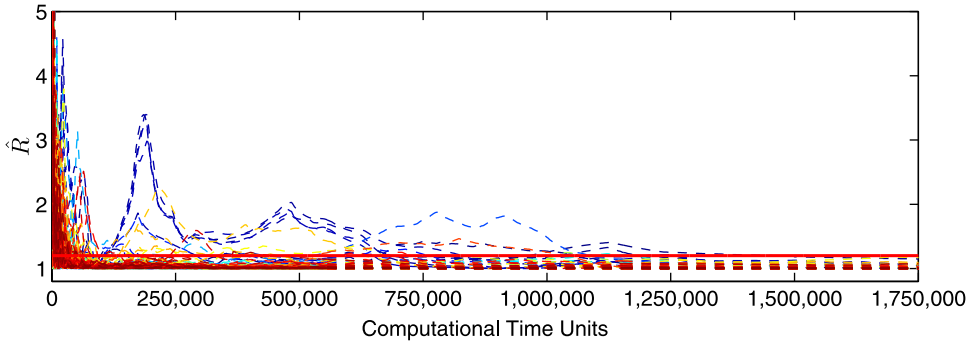

Figure 6. MT-DREAM$_{(ZS)}$: Evolution of the $\hat{R}$ statistic of *Gelman and Rubin* [1992] during the MT-DREAM$_{(ZS)}$ run for the 241-dimensional groundwater model calibration problem of *Keating et al.* [2010]. The different parameters $\hat{R}_j, j = 1, \ldots, d$ are coded with colored dashed lines. If the dashed lines jointly fall below the solid red line, convergence to a limiting distribution can be officially declared.

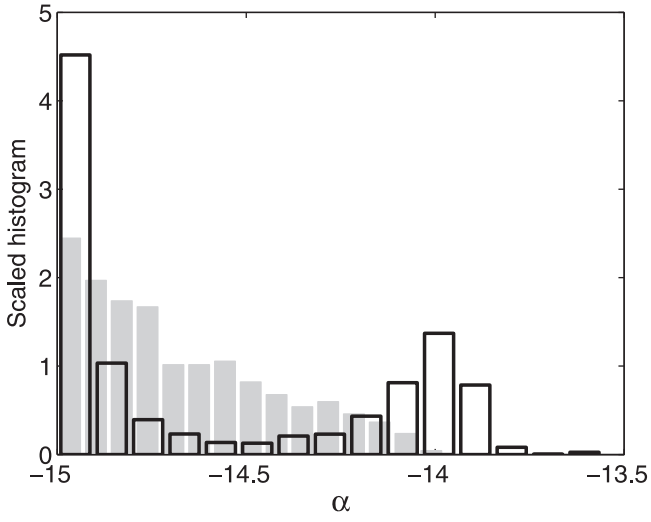

Figure 7. Histograms of the marginal posterior pdf of the α parameter in the groundwater model of *Keating et al.* [2010] derived with DREAM (gray bins) and MT-DREAM$_{(ZS)}$ (white bins). The marginal posterior distributions differ quite substantially, questioning some of the main conclusions of our previous work.

with DREAM. This is graphically illustrated in Figure 7 which compares the histogram of the marginal distribution of the parameter α [see *Keating et al.* 2010] sampled with DREAM (gray bins) and MT-DREAM$_{(ZS)}$ (white bins). Indeed, the marginal distributions deviate considerably from each other. This new insight brings into question our previous conclusions, and warrants additional analysis.

[63] Altogether, the results in this paper demonstrate some desirable advantages of multitry sampling for estimating complex, multimodal, and high-dimensional posterior distributions. The MT-DREAM$_{(ZS)}$ algorithm not only converges most rapidly from all the different MCMC samplers, but also provides samples that most closely represent the underlying target distribution. This superior performance might be somewhat surprising, considering that almost 90% of the samples (with $k = 5$) generated with MT-DREAM$_{(ZS)}$ are thrown away. This appears rather inefficient, yet, the summary statistics presented in Table 1, 2 and 4 illustrate that multitry sampling has some desirable advantages. Seemingly, 1000 high-quality and diverse samples created with MT-DREAM$_{(ZS)}$ contain more information about the posterior distribution than 10,000 lower-quality and less diverse samples generated with DREAM$_{(ZS)}$ and consisting of many different duplicates.

4. Conclusions

[64] Spatially distributed hydrologic models are increasingly being used to study the transport of water through catchments. These models contain a large number of parameters whose values cannot be measured directly in the field but can only meaningfully be obtained by calibration against a historical record of input-output data. The usefulness and applicability of such models thus essentially relies on the availability of powerful calibration methods that can efficiently summarize parameter and predictive uncertainty.

[65] In this paper, we have presented a novel MCMC algorithm entitled MT-DREAM$_{(ZS)}$ that is especially designed to solve high-dimensional model calibration problems and summarize posterior parameter distributions. This method combines the strengths of DREAM [*Vrugt et al.*, 2009], sampling from past states [*ter Braak and Vrugt*, 2008; Vrugt et al., manuscript in preparation, 2012], snooker updating [*ter Braak and Vrugt*, 2008], and multiple-try sampling [*Liu et al.*, 2000] to evolve an initial population of points to the posterior target distribution. MT-DREAM$_{(ZS)}$ is especially designed to be implemented on a distributed computing cluster. Detailed balance and ergodicity of the algorithm have been studied and ensured, which inspires confidence in the ability of MT-DREAM$_{(ZS)}$ to generate an example sample of the posterior distribution. Four different case studies with different peculiarities involving local optima, multimodality, and high parameter dimensionality, have shown that MT-DREAM$_{(ZS)}$ is generally superior to existing optimization and search approaches.

[66] There are various ways in which the efficiency of MT-DREAM$_{(ZS)}$ can be further improved, particularly for high-dimensional problems. For instance, one could use the ideas of *Tonkin and Doherty* [2005] and significantly reduce the dimensionality of the search space by reparameterizing the original inverse problem using super and base parameters derived from principal component analysis. This could significantly speed up the efficiency of posterior sampling, but our initial attempts to date have not been particularly successful, not only because each chain works in a different subspace (which makes it difficult to define the proposal distribution) but also because this approach violates detailed balance. More generally, coupling MT-DREAM$_{(ZS)}$ with dimensionality reduction techniques [e.g., *Marzouk and Najm* 2009] seems attractive to accelerate convergence to a limiting distribution. Another idea is to alternate the mix of parallel direction and snooker updates with a Langevin step [e.g., *Roberts and Rosenthal*, 1998] to sample proposal points preferentially in the direction of higher posterior density. This requires explicit knowledge of the gradient of $\pi(\cdot)$, which can be approximated from the archive of past states. We leave these ideas for future work.

Appendix A

[67] Convergence property of any type of MTM algorithm has been demonstrated by *Liu et al.* [2000]. The proof given here is similar. Before presenting the proof, it is important to recall that in the general MTM framework, we define

$$\omega(\mathbf{x}_{t-1}, \mathbf{z}) = \pi(\mathbf{x}_{t-1}) q(\mathbf{x}_{t-1}, \mathbf{z}) \lambda(\mathbf{x}_{t-1}, \mathbf{z}) \tag{A1}$$

where $\lambda(\mathbf{x}_{t-1}, \mathbf{z})$ is an user-defined nonnegative symmetric function in $\mathbf{x}_{t-1}$ and $\mathbf{z}$ ($\lambda(\mathbf{x}_{t-1}, \mathbf{z}) = \lambda(\mathbf{z}, \mathbf{x}_{t-1})$), which satisfies $\lambda(\mathbf{x}_{t-1}, \mathbf{z}) > 0$ whenever $q(\mathbf{x}_{t-1}, \mathbf{z}) > 0$. In the special case of MTM(II), we have $q(\mathbf{x}_{t-1}, \mathbf{z}) = q(\mathbf{z}, \mathbf{x}_{t-1})$ and set $\lambda(\mathbf{x}_{t-1}, \mathbf{z}) = q(\mathbf{z}, \mathbf{x}_{t-1})^{-1}$, which results in

$$\omega(\mathbf{x}_{t-1}, \mathbf{z}) = \pi(\mathbf{x}_{t-1}) \tag{A2}$$

Hence, the general MTM transition rule can be directly obtained by substituting $\pi(\cdot)$ by $\omega(\cdot)$ in the earlier description of the MTM(II) scheme (see equation (4)).

Theorem *The MTM transition rule satisfies the detailed balance condition and thus yields a reversible Markov chain with* $\pi(\cdot)^N$ *as its invariant distribution*.

Proof Recall that $p(\mathbf{x}_{t-1} \rightarrow \mathbf{z})$ is the transition probability for moving from $\mathbf{x}_{t-1}$ to $\mathbf{z}$. Assume that $\mathbf{x}_{t-1} \neq \mathbf{z}$ and let I indicate which $\mathbf{z}_j$ has been selected among the k proposal points, $j = 1, \ldots, k$. Because the $\mathbf{z}_j$ are exchangeable, we have

$$\begin{aligned}
\pi(\mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \rightarrow \mathbf{z}) &= \pi(\mathbf{x}_{t-1})P\left[\bigcup_{j=1}^{k}\{(\mathbf{z}_j = \mathbf{z}) \cap (I = j)\}|\mathbf{x}_{t-1}\right] \\
&= k\pi(\mathbf{x}_{t-1})P[(\mathbf{z}_k = \mathbf{z}) \cap (I = k)|\mathbf{x}_{t-1}] \\
&= k\pi(\mathbf{x}_{t-1}) \int \ldots \int q(\mathbf{x}_{t-1}, \mathbf{z})q(\mathbf{x}_{t-1}, \mathbf{z}_1) \ldots q(\mathbf{x}_{t-1}, \mathbf{z}_{k-1}) \\
&\times \frac{\omega(\mathbf{z}, \mathbf{x}_{t-1})}{\omega(\mathbf{z}, \mathbf{x}_{t-1}) \sum_{j=1}^{k-1} \omega(\mathbf{z}_j, \mathbf{x}_{t-1})} \\
&\times \min\left\{1, \frac{\omega(\mathbf{z}, \mathbf{x}_{t-1}) \sum_{j=1}^{k-1} \omega(\mathbf{z}_j, \mathbf{x}_{t-1})}{\omega(\mathbf{x}_{t-1}, \mathbf{z}) \sum_{j=1}^{k-1} \omega(\mathbf{x}_j^*, \mathbf{z})}\right\} \\
&\times q(\mathbf{z}, \mathbf{x}_1^*) \ldots q(\mathbf{z}, \mathbf{x}_{k-1}^*) d\mathbf{z}_1 \ldots d\mathbf{z}_{k-1} d\mathbf{x}_1^* \ldots d\mathbf{x}_{k-1}^* \\
&= k \frac{\omega(\mathbf{x}_{t-1}, \mathbf{z})\omega(\mathbf{z}, \mathbf{x}_{t-1})}{\lambda(\mathbf{z}, \mathbf{x}_{t-1})} \\
&\times \int \ldots \int \min\left\{\frac{1}{\omega(\mathbf{z}, \mathbf{x}_{t-1}) \sum_{j=1}^{k-1} \omega(\mathbf{z}_j, \mathbf{x}_{t-1})}, \frac{1}{\omega(\mathbf{x}_{t-1}, \mathbf{z}) \sum_{j=1}^{k-1} \omega(\mathbf{x}_j^*, \mathbf{z})}\right\} \\
&\times q(\mathbf{x}_{t-1}, \mathbf{z}_1) \ldots q(\mathbf{x}_{t-1}, \mathbf{z}_{k-1}) q(\mathbf{z}, \mathbf{x}_1^*) \ldots q(\mathbf{z}, \mathbf{x}_{k-1}^*) \\
&\times d\mathbf{z}_1 \ldots d\mathbf{z}_{k-1} d\mathbf{x}_1^* \ldots d\mathbf{x}_{k-1}^*
\end{aligned} \tag{A3}$$

[68] Because equation (A3) is symmetric in $\mathbf{x}_{t-1}$ and $\mathbf{z}$, $\pi(\mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \rightarrow \mathbf{z}) = \pi(\mathbf{z})p(\mathbf{z} \rightarrow \mathbf{x}_{t-1})$ which is the detailed balance condition.

[69] **Acknowledgments.** The authors thank three anonymous referee for their useful comments and suggestions. We acknowledge Elizabeth Keating for providing the surrogate groundwater model and associated calibration data. We also would like to thank Wei Chu for kindly sharing the SP-UCI source code. Computer support, provided by the SARA center for parallel computing at the University of Amsterdam, Netherlands, is highly appreciated.

References

Box, G. E. P., and G. C. Tiao (1973), *Bayesian Inference in Statistical Analysis*, Addison-Wesley-Longman, Reading, Mass.

Burnash, R. J. C. (1995), The NWS river forecast system-catchment modeling, in *Computer Models of Watershed Hydrology*, edited by V. P. Singh, pp. 311–366, Water Resour. Publ., Littleton, Colo.

Chib, S., F. Nardari, and N. Shepard (2002), Markov chain Monte Carlo methods for stochastic volatility models, *J. Econometrics*, *108*, 281–316.

Chu, W., X. Gao, and S. Sorooshian (2010), Improving the shuffled complex evolution scheme for optimization of complex nonlinear hydrological systems: Application to the calibration of the Sacramento soil moisture accounting model, *Water Resour. Res.*, *46*, W09530, doi:10.1029/2010WR009224.

Craiu, V. R., J. Rosenthal, and C. Yang (2009), Learn from thy neighbor: Parallel-chain and regional adaptive MCMC, *J. Am. Stat. Assoc*, *104*(488), 1454–1466.

Dekker, S. C., J. A. Vrugt, and R. J. Elkington (2012), Significant variation in vegetation characteristics and dynamics from ecohydrological optimality of net carbon profit, *Ecohydrology*, doi:10.1002/eco.177, in press.

Doherty, J. (2009), PEST: Model independent parameter estimation, software, Watermark Numer. Comput., Corinda, Queensland, Australia. [Available at http://www.pesthomepage.org.]

Duan, Q., V. K. Gupta, and S. Sorooshian (1992), Effective and efficient global optimization for conceptual rainfall-runoff models, *Water. Resour. Res*, *28*, 1015–1031.

El Adlouni, S., A.-C. Favre, and B. Bobée (2006), Comparison of methodologies to assess the convergence of Markov chain Monte Carlo methods, *Comput. Stat. Data Anal.*, *50*(10), 2685–2701.

Frenkel, D. (2004), Speed-up of Monte Carlo simulations by sampling of rejected states, *Proc. Natl. Acad. Sci. U. S. A.*, *101*(51), 457–472, doi:10.703/pnas.0407950101.

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

Geweke, J. (1992), Evaluating the accuracy of sampling-based approaches to calculating posterior moments, in *Bayesian Statistics* 4, edited by J. M. Bernado et al., pp. 169–194, Clarendon, Oxford, U. K.

Gilks, W. R., and G. O. Roberts (1996), Strategies for improving MCMC, in *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, pp. 89–114, Chapman and Hall, London.

Gilks, W. R., G. O. Roberts, and E. I. George (1994), Adaptive direction sampling, *Statistician*, *43*, 179–189.

Haario, H., E. Saksman, and J. Tamminen (1999), Adaptive proposal distribution for random walk Metropolis algorithm, *Comp. Stat.*, *14*(3), 375–395.

Haario, H., E. Saksman, and J. Tamminen (2001), An adaptive Metropolis algorithm, *Bernoulli*, *7*, 223–242.

Haario, H., E. Saksman, and J. Tamminen (2005), Componentwise adaptation for high dimensional MCMC, *Stat. Comput.*, *20*, 265–274.

Haario, H., M. Laine, A. Mira, and E. Saksman (2006), DRAM: Efficient adaptive MCMC, *Stat. Comp.*, *16*, 339–354.

Hansen, N., S. D. Muller, and P. Koumoutasakos (2003), Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMAES), *Evol. Comput.*, *11*(1), 1–18, doi:10.1162/106365603321828970.

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

Keating, E. H., J. Doherty, J. A. Vrugt, and Q. Kang (2010), Optimization and uncertainty assessment of strongly nonlinear groundwater models with high parameter dimensionality, *Water Resour. Res.*, *46*, W10517, doi:10.1029/2009WR008584.

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.

Laloy, E., D. Fasbender, and C. L. Bielders (2010a), Parameter optimization and uncertainty analysis for plot-scale continuous modeling of runoff using a formal Bayesian approach, *J. Hydrol.*, *380*(1–2), 82–93, doi:10.1016/j.jhydrol.2009.10.025.

Laloy, E., M. Weynants, C. L. Bielders, M. Vanclooster, and M. Javaux (2010b), How efficient are one-dimensional models to reproduce the hydrodynamic behavior of structured soils subjected to multi-step outflow experiments? *J. Hydrol.*, *393*(1–2), 37–52, doi:10.1016/j.jhydrol.2010.02.017.

Liu, J. S., F. Liang, and W. H. Wong (2000), The multiple-try method and local optimization in Metropolis sampling, *J. Am. Stat. Assoc.*, *95*(449), 121–134, doi:10.2307/2669532.

Marquardt, D. W. (1963), An algorithm for least-squares estimation of nonlinear parameters, *J. Soc. Ind. Appl. Math.*, *11*, 431–441.

Marzouk, Y. M., and H. N. Najm (2009), Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems, *J. Comput. Phys.*, *118*, 862–902, doi:10.1016/j.jcp.2008.11.024.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953), Equation of state calculations by fast computing machines, *J. Chem. Phys.*, *21*, 1087–1092.

Murray, I. (2007), Advances in Markov chain Monte Carlo methods, PhD thesis, Univ. of London, London.

Raftery, A. E., and S. M. Lewis (1992), One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo, *Stat. Sci.*, *7*, 493–497.

Roberts, G. O., and W. R. Gilks (1994), Convergence of adaptive direction sampling, *J. Multivariate Anal.*, *49*, 287–298.

Roberts, G. O., and J. S. Rosenthal (1998), Optimal scaling of discrete approximations to Langevin diffusions, *J. R. Stat. Soc., Ser. B*, *60*, 255–268.

Roberts, G. O., and J. S. Rosenthal (2007), Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms, *J. Appl. Probab.*, *44*, 458–475.

Roberts, G. O., A. Gelman, and W. R. Gilks (1997), Weak convergence and optimal scaling of random walk Metropolis algorithms, *Ann. Appl. Probab.*, *7*, 110–120.

Scharnagl, B., J. A. Vrugt, H. Vereecken, and M. Herbst (2010), Information content of incubation experiments for inverse modeling of carbon pools in the Rothamsted model: A Bayesian approach, *Biogeosciences*, *7*, 763–776.

Schoups, G., and J. A. Vrugt (2010), A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic and non-Gaussian errors, *Water Resour. Res.*, *46*, W10531, doi:10.1029/2009WR008933.

ter Braak, C. J. F. (2006), A Markov chain Monte Carlo version of the genetic algorithm differential evolution: Easy Bayesian computing for real parameter space, *Stat. Comput.*, *16*(3), 239–249, doi:10.1007/s11222-006-8769-1.

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, doi:10.1007/s11222-008-9104-9.

Thiemann, M., M. Trosset, H. Gupta, and S. Sorooshian (2001), Bayesian recursive parameter estimation for hydrologic models, *Water Resour. Res.*, *37*(10), 2521–2535, doi:10.1029/2000WR900405.

Tonkin, M., and J. Doherty (2005), A hybrid regularized inversion methodology for highly parameterized environmental models, *Water Resour. Res.*, *41*, W10412, doi:10.1029/2005WR003995.

Tonkin, M., and J. Doherty (2009), Calibration constrained Monte Carlo analysis of highly parameterized models using subspace techniques, *Water Resour. Res.*, *45*, W00B10, doi:10.1029/2007WR006678.

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*(14), 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, D. Higdon, B. A. Robinson, and J. M. Hyman (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.

Yapo, P. O., H. V. Gupta, and S. Sorooshian (1996), Calibration of conceptual rainfall-runoff models: Sensitivity to calibration data, *J. Hydrol.*, *181*, 23–48.

E. Laloy and J. A. Vrugt, Department of Civil and Environmental Engineering, University of California, Irvine, CA 92697, USA. (elaloy@uci.edu)