Environmental Modelling & Software 60 (2014) 45–56

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

A computer program for uncertainty analysis integrating regression and Bayesian methods

Dan Lu [a, *], Ming Ye [b], Mary C. Hill [c], Eileen P. Poeter [d], Gary P. Curtis [e]

[a] Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[b] Department of Scientific Computing, Florida State University, Tallahassee, FL 32306, USA
[c] U.S. Geological Survey, Boulder, CO 80303, USA
[d] Integrated Ground Water Modeling Center, Department of Geology and Geological Engineering, Colorado School of Mines, Golden, CO 80401, USA
[e] U.S. Geological Survey, Menlo Park, CA 94025, USA

ARTICLE INFO

Article history:
Received 18 November 2013
Received in revised form
1 June 2014
Accepted 2 June 2014
Available online

Keywords:
Markov Chain Monte Carlo
UCODE_2014
Bayesian uncertainty analysis

ABSTRACT

This work develops a new functionality in UCODE_2014 to evaluate Bayesian credible intervals using the Markov Chain Monte Carlo (MCMC) method. The MCMC capability in UCODE_2014 is based on the FORTRAN version of the differential evolution adaptive Metropolis (DREAM) algorithm of Vrugt et al. (2009), which estimates the posterior probability density function of model parameters in high-dimensional and multimodal sampling problems. The UCODE MCMC capability provides eleven prior probability distributions and three ways to initialize the sampling process. It evaluates parametric and predictive uncertainties and it has parallel computing capability based on multiple chains to accelerate the sampling process. This paper tests and demonstrates the MCMC capability using a 10-dimensional multimodal mathematical function, a 100-dimensional Gaussian function, and a groundwater reactive transport model. The use of the MCMC capability is made straightforward and flexible by adopting the JUPITER API protocol. With the new MCMC capability, UCODE_2014 can be used to calculate three types of uncertainty intervals, which all can account for prior information: (1) linear confidence intervals which require linearity and Gaussian error assumptions and typically 10s–100s of highly parallelizable model runs after optimization, (2) nonlinear confidence intervals which require a smooth objective function surface and Gaussian observation error assumptions and typically 100s–1,000s of partially parallelizable model runs after optimization, and (3) MCMC Bayesian credible intervals which require few assumptions and commonly 10,000s–100,000s or more partially parallelizable model runs. Ready access allows users to select methods best suited to their work, and to compare methods in many circumstances.

Published by Elsevier Ltd.

Software availability

Name of software: The Markov Chain Monte Carlo capability in UCODE_2014
Description: The Markov Chain Monte Carlo capability developed in UCODE_2014 to generate parameter samples and evaluate parametric and predictive Bayesian uncertainties
Developer: Dan Lu (lud1@ornl.gov), Mary Hill (mchill@usgs.gov), and Eileen Poeter (epoeter@mines.edu)
Programming language: Fortran
Availability: Download from website http://igwmc.mines.edu/freeware/ucode/

1. Introduction

Quantifying uncertainty in evaluations and predictions of how anthropogenic and/or natural events affect the environment is an important step of any mathematically based modeling effort. The new version of UCODE, UCODE_2014, provides a set of uncertainty quantification methods that range from computationally frugal regression methods (as few as 10s–100s of model runs after optimization) with often significant restrictive assumptions, to computationally demanding Bayesian methods (commonly 10,000s–100,000s of model runs) with few restrictive assumptions. All methods are able to account for prior information. Having

* Corresponding author.
E-mail address: lud1@ornl.gov (D. Lu).

http://dx.doi.org/10.1016/j.envsoft.2014.06.002
1364-8152/Published by Elsevier Ltd.

this range of methods readily accessible to users as provided by UCODE_2014 is important to the following goals:

(1) investigative studies in which the different uncertainty intervals types are compared and guidance is provided about circumstances for which the more computationally expensive Bayesian credible intervals are likely to be important and when the computationally cheaper regression confidence intervals are potentially useful (for example, see Lu et al. (2012)),
(2) progressive calculation of intervals so that computationally frugal regression confidence intervals can be used routinely earlier in a study while the more expensive Bayesian credible intervals can be calculated occasionally and often later in the study,
(3) calculation of only Bayesian credible intervals (as needed for models with very irregular objective function surfaces and often with multiple local minima) and,
(4) calculation of only computationally frugal regression confidence intervals (as needed to enable use of computationally demanding models and evaluation using multiple alternative models, and valid if linearity or smoothness, and Gaussian assumptions are not violated too much).

A program supporting such flexible strategies is needed because of limitations in the existing programs developed for uncertainty analysis in the environmental community. For example, the DAKOTA optimization and uncertainty software (Adams et al., 2013) previously evaluated the Bayesian credible intervals using the DRAM algorithm (Haario et al., 2006) which can be less efficient and unreliable for complex and multimodal problems than the DREAM algorithm used in UCODE_2014 (Vrugt et al., 2009). DREAM is now in the process of being implemented in DAKOTA. UCODE_2005 (Poeter et al., 2005) and PEST (Doherty, 2005) (which are both inverse modeling codes that can be used with any process models with ASCII-based inputs and outputs) provide uncertainty analysis with linear and nonlinear regression confidence intervals. Null space Monte Carlo (NSMC), another uncertainty analysis method encapsulated in PEST, provides predictive probability distributions in a computationally efficient way (Keating et al., 2010), but can display erratic performance (Laloy and Vrugt, 2012). iTOUGH2 (Finsterle and Zhang, 2011a,b) evaluates predictive uncertainty using linear uncertainty propagation and simple Monte Carlo analysis based on the distributions of uncertain parameters. Although they both use a Monte Carlo method, neither NSMC nor iTOUGH2 analyzes the predictive uncertainty in a rigorous Bayesian way by evaluating the posterior distributions. MICA (Doherty, 2003) and DREAM (Vrugt et al., 2008, 2009) (which are both MCMC codes that can be used to generate parameter samples from their posterior probability distribution) calculate only Bayesian credible intervals. None of the listed programs calculate both regression confidence intervals and Bayesian credible intervals efficiently.

MICA and DREAM are the two most widely used programs in the environmental community for Bayesian uncertainty analysis, and both codes are available at no charge from the developers. MICA is developed based on the Metropolis–Hastings algorithm. It is easy and straightforward to use with any process model that uses ASCII-based inputs and outputs. MICA input file and template and instruction files are similar or equivalent to those of PEST; the template and instruction files can also be used with UCODE_2014. MICA provides a wide range of probability density functions for the MCMC parameter prior distribution, and can evaluate parametric uncertainties for any or all model parameters and derived parameters. MICA cannot perform parallel MCMC computations. MICA works well for estimating unimodal posterior distributions, but for multimodal problems it cannot sample the target posterior distribution efficiently with a single proposal distribution (Gallagher and Doherty, 2007; Lu et al., 2012). This problem is resolved by DREAM (Vrugt et al., 2008, 2009) which is described in Section 2 of this paper.

This work integrates the DREAM algorithm into UCODE_2014, which is documented by Poeter et al. (2005, 2014). Inspired by the structure of MICA, the UCODE_2014 MCMC capability is user-friendly and can be easily used without in-depth knowledge of MCMC. The MCMC capability generates parameter samples and produces Bayesian predictive uncertainty by calculating model predictions from the generated parameter samples after a burn-in period (i.e., the parameter samples after chain convergence). The MCMC simulation in UCODE_2014 has parallel computing capability where the process model runs for different chains are accomplished on different processors for simultaneous execution. This greatly accelerates the MCMC sampling process.

With the new MCMC capability, UCODE_2014 can be used to calculate three types of uncertainty intervals: linear and nonlinear confidence intervals and Bayesian credible intervals. Confidence intervals are based on regression theories and credible intervals are based on Bayesian theories. While both can include the effect of prior information, confidence and credible intervals are conceptually different, and their differences and similarities are discussed in statistical literature including Jaynes (1976), Bates and Watts (1988), and Box and Tiao (1992). A recent discussion and literature review in the context of environmental modeling is presented by Lu et al. (2012). Given a nonlinear model and multi-Gaussian distributed observation errors, theory suggests that nonlinear confidence and credible intervals can be numerically identical if model nonlinearity is "small enough" and there are no local minima. They present a groundwater flow problem which indicates that even linear intervals can provide useful evaluations of uncertainty given common levels of nonlinearity. However, many environmental problems are so nonlinear that Bayesian methods with less restrictive assumptions are needed, and the ability to calculate both regression confidence intervals and Bayesian credible intervals is important (Vrugt and Bouten, 2002; Gallagher and Doherty, 2007; Liu et al., 2010; Shi et al., 2012, 2014).

The computational cost of calculating the confidence and credible intervals can be considerably different. Calculating the linear and nonlinear confidence intervals typically requires 10s–1,000s of model runs after a calibrated model is achieved. Model calibration includes identifying both the best fit parameter values and other aspects of model development. For a given model, one MCMC simulation can determine both the best fit parameter values and credible intervals that require neither smoothness nor Gaussian error assumptions. MCMC credible intervals can sometimes be obtained using 1,000s of model runs, but commonly require 10,000s, and even millions of model runs. For all methods, the number of model runs required tends to increase with problem dimensionality, though with linear confidence intervals the rate of increase is plus two runs for each additional parameter, more for nonlinear confidence intervals, and much more for MCMC credible intervals. Increasing nonlinearity leads to more model runs for nonlinear confidence intervals and MCMC credible intervals.

This paper introduces the MCMC capability implemented in UCODE_2014 and presents extensive tests. First, the MCMC method is briefly described in Section 2 with emphasis on the UCODE_2014 implementation. In Section 3, the features of the capability are discussed in detail. In Section 4, a 10-dimensional multimodal mathematical function and a 100-dimensional Gaussian function are used to test the MCMC capability in complex sampling problems, and a groundwater reactive transport model is presented to

illustrate that it is straightforward and flexible to use. Last, concluding remarks are summarized in Section 5.

2. MCMC DREAM algorithm in UCODE_2014

2.1. MCMC method and DREAM algorithm

Bayesian analysis estimates the posterior probability density functions of model parameters and predictions. Based on the Bayes' theorem, the posterior distribution $p(\boldsymbol{\theta}|\mathbf{D})$ of model parameters $\boldsymbol{\theta}$ given observation data $\mathbf{D}$ combines the data likelihood $L(\boldsymbol{\theta}|\mathbf{D}) = p(\mathbf{D}|\boldsymbol{\theta})$ with the parameter prior distribution $p(\boldsymbol{\theta})$, as follows,

$$p(\boldsymbol{\theta}|\mathbf{D}) = L(\boldsymbol{\theta}|\mathbf{D})p(\boldsymbol{\theta}) \Big/ \int L(\boldsymbol{\theta}|\mathbf{D})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \quad (1)$$

Assuming multi-Gaussian distributed errors in the observations, the likelihood function is constructed as,

$$\begin{aligned} L(\boldsymbol{\theta}|\mathbf{D}) &= p(\mathbf{D}|\boldsymbol{\theta}) \\ &= (2\pi)^{-N/2}|\mathbf{C}|^{-1/2}\exp\left(-\frac{1}{2}\left(\mathbf{D}-\widehat{\mathbf{D}}\right)^T\mathbf{C}^{-1}\left(\mathbf{D}-\widehat{\mathbf{D}}\right)\right) \end{aligned} \quad (2)$$

where N is the number of data, $\mathbf{C}$ is the covariance matrix of observation errors, $\widehat{\mathbf{D}}$ is a vector of simulated values, and term $(\mathbf{D}-\widehat{\mathbf{D}})^T\mathbf{C}^{-1}(\mathbf{D}-\widehat{\mathbf{D}})$ represents the sum of squared weighted residuals (SSWR). Bayesian analysis supports different kinds of likelihood functions, and is not limited to the Gaussian function used in UCODE_2014 (Smith et al., 2010; Zhang et al., 2013).

A number of MCMC techniques have been developed for Bayesian analysis in environmental modeling. Most use the Metropolis sampler (Metropolis et al., 1953) and differ from each other mainly in construction of the proposal distribution to evolve the chain. In the Metropolis algorithm, after initializing the chain by drawing a parameter sample $\boldsymbol{\theta}^t$ from a certain distribution (e.g., the parameter prior distribution) the sampler proceeds in the following three steps. First, a candidate sample $\boldsymbol{\theta}^*$ is sampled from a proposal distribution q, which is required to be symmetric, $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^t) = q(\boldsymbol{\theta}^t|\boldsymbol{\theta}^*)$. Next, the candidate sample is either accepted or rejected based on the following Metropolis ratio α:

$$\alpha = \frac{p(\boldsymbol{\theta}^*|\mathbf{D})}{p(\boldsymbol{\theta}^t|\mathbf{D})} \begin{cases} \geq 1, \boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^* \\ < 1 \begin{cases} \alpha > r \sim U(0,1), \ \boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^* \\ \text{otherwise}, \ \boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t \end{cases} \end{cases} \quad (3)$$

where $p(\boldsymbol{\theta}^*|\mathbf{D})$ is calculated with equation (1) in which the likelihood $L(\boldsymbol{\theta}^*|\mathbf{D})$ is calculated for $\boldsymbol{\theta}^*$ using equation (2), and the calculation of $p(\boldsymbol{\theta}^t|\mathbf{D})$ is conducted in the same manner. In equation (3), the denominator of equation (1) cancels, thus avoiding the computationally expensive multiple dimensional integration. Finally, if the candidate sample is accepted the chain uses $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^*$ as the sample at iteration $t+1$; otherwise the chain keeps using the current sample $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t$.

Theory shows that the Metropolis sampler converges to the target distribution $p(\boldsymbol{\theta}|\mathbf{D})$ under certain regularity conditions (Robert and Casella, 2004, p.270). In practice, the convergence is often observed to be impractically slow due to problems with the selected proposal distribution. When the proposal distribution is too wide, very few candidate samples are accepted, and the chain does not mix properly and thus converges slowly. If the proposal distribution is too narrow, the acceptance rate is rather large, and the chain does not reach far enough and thus it requires many iterations to sample the entire posterior distribution. Among the existing MCMC algorithms, DREAM is able to automatically tune the scale and orientation of the proposal distribution in randomized parameter subspaces during its global exploration. This significantly enhances the computational efficiency for complex and multimodal problems. DREAM accelerates the convergence in three ways. First, DREAM only updates selected dimensions when a candidate sample is generated, which improves efficiency for high-dimensional problems because with increasing dimensions it is often not optimal to change all dimensions simultaneously. Second, DREAM generates the candidate sample using a differential evolution offspring strategy with use of more than one pair of chains, which increases the diversity of the proposal and thus facilitates the updating of the proposal distribution to the target. Third, DREAM explicitly handles and removes chains stuck in nonproductive parts of the parameter space, which speeds up convergence to the target distribution. More information about the DREAM algorithm is presented in Vrugt et al. (2008, 2009).

MCMC convergence can be diagnosed by the Gelman-Rubin $\widehat{R}$ statistic calculated from multiple chains running simultaneously (Gelman et al., 1995). $\widehat{R}$ below 1.2 is considered as a practical criterion of convergence by Gelman and Rubin (1992). After convergence, the generated parameter samples are taken from the target distribution $p(\boldsymbol{\theta}|\mathbf{D})$ and can be used for Bayesian uncertainty analysis to compute the credible intervals. The threshold of 1.2 is a rule of thumb and the samples from later iterations represent the target distribution more accurately, so in practice when the number of parameter samples after chain convergence is larger than the 50% of the total samples in each chain, the last 50% of the samples in the joint chains are commonly used for the uncertainty analysis (Gelman and Rubin, 1992; Laloy and Vrugt, 2012).

2.2. Existing MCMC DREAM programs and relation to UCODE_2014

The original version of DREAM is written in MATLAB. For those who are familiar with MATLAB, a simulation can be easily set up by specifying certain variables in the input file. For those who have limited programming expertise, the DREAM MATLAB code provides versatile examples which should be sufficient for them to set up a new problem. The original MATLAB version of DREAM has been recently translated to different languages such as C, C++, FORTRAN, R, and Python, and can be downloaded from http://faculty.sites.uci.edu/jasper/sample/.

Parallelization of the DREAM MATLAB code is implemented in Octave using the MPITB toolbox. With this approach, only minor modifications are needed to the original sequential MATLAB source code to facilitate implementation on a distributed computer system. Implementation of the parallel DREAM MATLAB code can proceed as shown in the examples presented in Vrugt et al. (2006) and Laloy and Vrugt (2012). The parallel R version of DREAM can be set up as described by Joseph and Guillaume (2013).

For UCODE_2014, the MCMC DREAM algorithm was written in FORTRAN using capabilities of the JUPITER API (Banta et al., 2006, 2008) to enable user-friendly data input, UCODE- and PEST-compatible template and instruction files, and efficient access to multiple processors for parallelization. The MCMC functions developed in UCODE_2014 are largely a migration of a subset of existing functions implemented in the series of DREAM codes developed by Jasper Vrugt; the MCMC capabilities are not advanced as part of this work and it only considers Gaussian likelihood function. To serve the goals discussed in the introduction of this paper, UCODE_2014 provides unique access to uncertainty quantification methods that range from very computationally frugal linear regression method to the computationally demanding MCMC Bayesian method.

2.3. UCODE_2014 structures and features

Previous versions of UCODE (most recently, UCODE_2005) have been widely used in environmental modeling for sensitivity analysis, data needs assessment, calibration, prediction, and uncertainty analysis. All these analyses are based on regression theory, which makes the integration of MCMC into UCODE_2014 for Bayesian uncertainty analysis necessary. UCODE_2014 (like previous versions of UCODE) is constructed based on the JUPITER API (Banta et al., 2006, 2008) whose structures are designed to support development of computer programs dedicated to model analysis. Their utility in this integration of MCMC into UCODE_2014 includes the following.

1. **Interacting with any process model in an easy and flexible way.** UCODE_2014 first uses a template file to create process model input files with provided parameter values; then it uses command line in batch mode to execute the process model(s); last it uses an instruction file to extract simulated values from the process model output file. This procedure and files follow the same conventions as PEST, so PEST applications can easily take advantage of the MCMC capability in UCODE_2014, and UCODE_2014 applications can easily take advantage of PEST capabilities.
2. **Flexible input design based on input blocks with keywords.** Unique keywords in the UCODE_2014 main input file control specific capabilities of UCODE_2014. For example, in UCODE_2014, MCMC simulation can be activated by setting the keyword MCMC as yes. The employment of input blocks facilitates reuse of constructed input in different applications. For example, the observation and prediction input blocks used to analyze the regression predictive uncertainty can also be used for MCMC Bayesian uncertainty analysis.
3. **Data-exchange files for improved communication between applications.** For example, the regression results saved in data-exchange files can be used to initialize the MCMC process as discussed below.
4. **The flexible equation capability.** This can be used to analyze uncertainty for predictions which are functions of model outputs, e.g., streamflow loss and gains.

In addition, UCODE_2014 is capable of calculating measures of model nonlinearity (e.g., using Beales nonlinearity measure based on calibration results) which can be used for preliminary evaluation about how close regression confidence intervals and Bayesian credible intervals are likely to be (Poeter et al., 2005; Lu et al., 2012). For problems with large measures of model nonlinearity, uncertainty will be evaluated more accurately using Bayesian credible intervals. Problems with smaller measures of model nonlinearity can have local minima which again are better evaluated using Bayesian credible intervals. If credible intervals are to be calculated, the above outlined progressive calculation of confidence and credible intervals is useful and the capabilities of UCODE_2014 make this easy. The switch from one calculation to the other can be implemented by simply changing several keywords and input blocks in the main input file.

2.4. Procedure of the MCMC capability in UCODE_2014

The MCMC procedure in UCODE_2014 is presented in Fig. 1. It consists of two separate runs, generating parameter samples to evaluate parametric uncertainty and generating prediction samples to evaluate predictive uncertainty. The two runs are conducted sequentially. In generating parameter samples, for one step of a sequential run, the program first generates a parameter sample (a vector of all parameters) for the first chain, then moves to another chain. After going through all the chains, it repeats the process to generate the next sample for the first chain, and so on. For a parallel run, the independent process model runs of different chains are assigned to different processors for simultaneous execution. In generating prediction samples, the converged parameter samples from all chains are evaluated. This process can be run sequentially or fully parallelized.

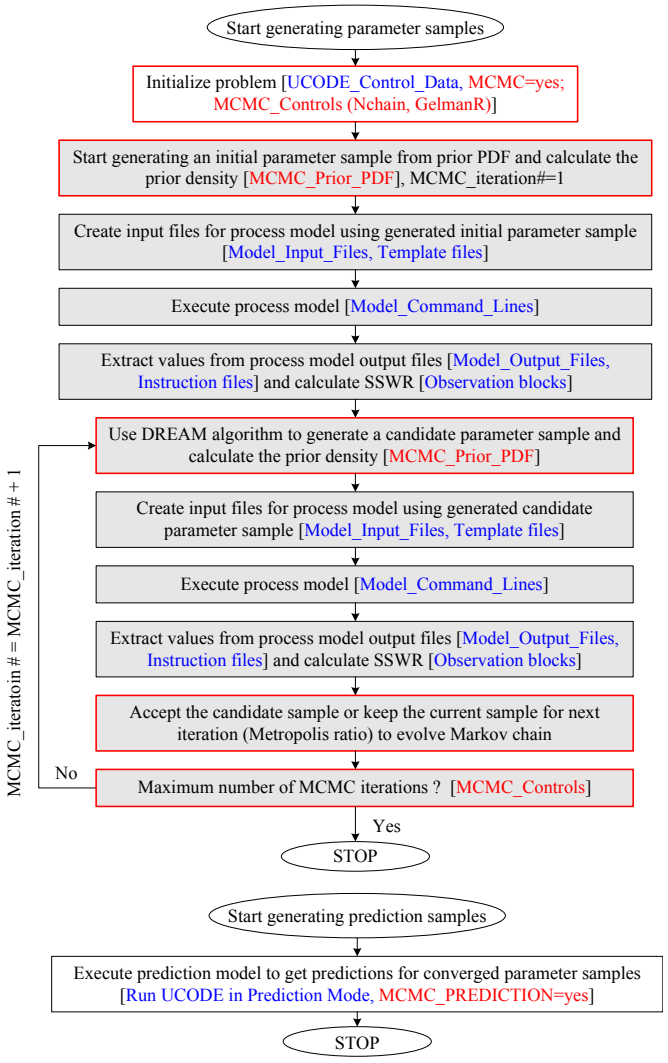

Fig. 1. Flowchart of MCMC capability in UCODE_2014 with the MCMC part shown in red and UCODE_2014 input blocks other than those unique to MCMC shown in blue. Input blocks with keywords that control the listed performance are listed in brackets. The boxes with gray shading are repeated in each chain. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In the entire implementation, the core of the MCMC process remains intact, i.e., its initialization of the chains, its generation of the candidate samples, and its use of the Metropolis ratio to evolve the chains. UCODE_2014 capabilities are used to (1) transfer parameter values to process model input files, (2) execute the process model(s), and (3) assemble output values. Future enhancements to the MCMC algorithm will not affect the simulation setup, because the modularized program structure allows all other aspects of UCODE_2014 to remain the same.

3. Features of the MCMC capability in UCODE_2014

The MCMC capability in UCODE_2014 has features not only related to the advanced algorithm of MCMC sampler (DREAM) but also of the UCODE structures as mentioned above. Some of these features are not available or are available in a substantially less convenient manner in other MCMC programs for the application of Bayesian uncertainty analysis. In this section, major features of the MCMC capability are discussed in detail.

3.1. Initializing MCMC simulation in multiple ways

How Markov chains are started affects simulation efficiency. In UCODE_2014, Markov chains can be started in the following three ways:

(1) Each chain is started with the first parameter sample drawn from the parameter prior distributions defined in the MCMC_Prior_PDF input block. This is the most general way to start a chain if little information of the target posterior distribution is available.
(2) Each chain is started with the first parameter sample drawn from a multi-Gaussian distribution with mean and covariance matrix from UCODE_2014 parameter optimization analysis. In this work, this initialization of chains is shown to be efficient for unimodal problems. For a multimodal problem, it may take more iterations to reach convergence than option (1).
(3) If a previous MCMC run was conducted, but either it did not converge or the number of generated parameter samples was determined to be insufficient, then a new MCMC run can be restarted where the previous run terminated. With this initialization, much computational time can be saved.

3.2. Parameter prior distributions

Parameter prior distribution plays an important role in MCMC simulation for Bayesian uncertainty analysis. The prior distribution can be used to initialize the Markov chains. The prior density needs to be calculated when evaluating the Metropolis ratio (equation (3)). When the prior distribution is informative and the number of calibration data is not large, the prior distribution can greatly affect the target posterior distribution. UCODE_2014 provides eleven widely used distributions, ten of which are for individual parameters and one for interdependent parameters. The ten univariate distributions are uniform, normal, log-normal, beta, chi-square, inverse chi-square, scaled inverse chi-square, gamma, inverse gamma, and exponential. Currently, only the multivariate normal distribution is available for considering interdependence between parameters.

3.3. Accounting for observation errors

In the calculation of SSWR values, UCODE_2014 (like previous versions of UCODE) allows weighting to be calculated as the inverse of a variance/covariance matrix of observation errors (Hill and Tiedeman, 2007). UCODE_2014 has the following three design features in regard to accounting for the observation errors.

(1) If the errors are considered to be independent, the weighting is a diagonal matrix with the weight for each observation equal to $1/\sigma_i^2$, where σ_i^2 is the variance of the associated error. Besides the variance, UCODE_2014 conveniently allows users to provide a standard deviation or coefficient of variation (CV), and it then calculates the weights internally. If CV is used, UCODE_2014 provides enhanced capabilities for calculating the weights using observed or simulated values, as considered by Anderman and Hill (1999).
(2) If the errors are taken to be correlated, UCODE_2014 allows users to specify a full covariance matrix of the errors, and it then calculates the inverse of the matrix internally to obtain the weights.
(3) The third feature is a special situation of feature (2). UCODE_2014 allows users to input the full covariance matrix in a compressed format to reduce the computer storage for what is often a large and sparse matrix. For more information see Poeter et al. (2005).

3.4. Structured inputs and outputs

An MCMC simulation in UCODE_2014 needs the main input file for simulation setup, and the template and instruction files for communication with the process model. In some special situations, extra inputs may be required. For example, when the MCMC simulation starts with the parameter sample drawn from a multi-Gaussian distribution defined with UCODE_2014 parameter optimization results, the simulation needs the files in which UCODE_2014 saves optimal parameter estimates and their covariance matrix (i.e., data-exchange files _paopt and _mv).

The MCMC outputs include the main output file that contains echoes of the inputs and the iteration number at which chains converge based on the $\widehat{R}$ statistic, and data-exchange files. For example, the parameter samples and associated simulated values and log likelihood functions at a certain iteration interval for each chain are written in files that can be opened by common software such as Excel for post-processing. The iteration interval is specified by users in the MCMC_Controls input block using the keyword PrintStep; with a large value of PrintStep the size of the output files can be reduced. By plotting the samples for all chains versus iteration number using common software such as Excel, four characteristics are revealed: (1) the chains' evolutions to the target posterior distribution, (2) a rough estimate of whether and when the chains are mixed, (3) whether the parameter posterior distribution has multiple modes, and (4) how the chains jump between the modes. Moreover, by plotting the sample mean, sample standard deviation and sample covariance along with the number of iterations, one can diagnose chain convergence visually. With sufficient parameter samples after chain convergence, the last 50% of the samples in all chains are used to plot the histogram of each individual parameter to estimate its marginal posterior distribution and to construct the scatter plots of any two parameters to analyze parameter correlation. The parametric uncertainties can be evaluated by calculating the equal-tailed credible intervals. Based on the output files of simulated values, the normality of the residuals can be evaluated to test the Gaussian assumption. The output files also include Gelman-Rubin $\widehat{R}$ information at an iteration interval specified by users with the same keyword PrintStep as mentioned above. Finally the last parameter sample and associated log likelihood function for all chains, which are needed to restart the chains, are written in a file. These output files can be imported to GW-Chart (http://water.usgs.gov/nrp/gwsoftware/GW_Chart/GW_Chart.html) for statistical analysis.

3.5. Predictive uncertainty analysis

Analyzing Bayesian predictive uncertainty is accomplished using a separate run of UCODE_2014. In this run, the main input file

for the MCMC run can be used with the following changes: add the prediction input block and change the forward model run to a prediction model run. The prediction quantities can be model outputs or functions of model outputs. After activating the MCMC prediction run, UCODE_2014 evaluates the predictions for a sequence of parameter samples starting with sample ItStartPred, which is defined by the user in the MCMC_Controls input block. UCODE_2014 produces a file recording the prediction samples for each chain. The predictive uncertainty can be assessed in the same manner as discussed in Section 3.4 for parametric uncertainty.

3.6. Parallel computing capability for MCMC simulation

In the MCMC run, for each generated parameter sample, the process model needs to be executed. The number of samples is generally large, often exceeding 10,000. For each chain each run depends on the preceding run, so parallelization is not possible within a chain. However, when the MCMC simulation is performed by launching several chains, parallelization opportunities exist.

To run the program in parallel, the only requirement is network read-and-write access between processors. This can be implemented in both Windows and Linux operating systems. In principle, DREAM requires at least three chains for the differential evolution offspring strategy to generate a candidate sample, and at least $N_{\text{chain}} = N_p/2$ to N_p chains (dependent on problem complexity) for parallel runs to be efficient, where N_{chain} is the number of chains and N_p is the number of parameters. In situations with limited processors and a moderate number of parameters, it is suggested that N_{chain} equal N_p for computational efficiency (fewer chains imply fewer samples discarded in the burn-in period) and for approximation accuracy (more chains imply more possible areas of the posterior distribution that could be efficiently sampled). If enough processors are available, more chains than the number of parameters can be used. Ideally, this can achieve speedups nearly proportional to the number of processors available and in our experience the speedups of about half the number of processors are more common. This greatly reduces the execution time and makes the routine use of Bayesian uncertainty analysis possible for many problems. For complex problems with many local minima, it is suggested that N_{chain} be greater than N_p to increase the diversity of the proposals and accelerate the visits of all possible modes.

In the MCMC prediction run, for each converged parameter sample, the predictions need to be evaluated. This process can be fully parallelized since all the samples are independent and the user can use as many processors as possible to achieve the largest efficiency.

4. Applications of the MCMC capability in UCODE_2014

In this section, we demonstrate the features and applicability of the MCMC capability in UCODE_2014 for three case studies with an increasing level of complexity. The first two case studies are mathematical functions, one has 10 dimensions with two modes and the other has 100 dimensions. They are designed to test the MCMC capability to infer a known posterior target distribution for multimodal and high-dimensional problems. The third case study considers an example in groundwater reactive transport modeling to demonstrate potential applications of the program. The discussion focuses on the program's features rather than on the scientific contents of the analysis.

4.1. Case study I: a 10-dimensional bimodal mathematical function

This case study is used to test the MCMC DREAM algorithm implementation in UCODE_2014 by simulating a known posterior probability density function with multimodality. A 10-dimensional (10 parameters) bimodal function for which each parameter has two well-separated modes was adapted from Ter Braak (2006) and Vrugt et al. (2009). It has the form

$$p(\boldsymbol{\theta}) = 1/3N_{10}(-\mathbf{5}, \mathbf{I}_{10}) + 2/3N_{10}(\mathbf{5}, \mathbf{I}_{10}) \quad (4)$$

where $N_{10}(-\mathbf{5},\mathbf{I}_{10})$ represents the 10-dimensional multivariate normal distribution with mean $-\mathbf{5}$ and covariance of identity matrix, and $N_{10}(\mathbf{5},\mathbf{I}_{10})$ represents the same distribution but with mean $\mathbf{5}$. The distribution has two separated modes at -5 and 5 in each dimension of the 10-dimensional space. This type of distribution is known for its difficulty to approximate with MCMC, because the chains need to jump from one mode to the other and it is common for chains to get stuck in one mode with a few or even no visits to the other mode for many iterations. This greatly increases the number of iterations to achieve convergence.

Ten chains are launched by drawing the first sample from a uniform distribution within the range -20 to 20 for each of the ten parameters. Each chain evolves 50,000 iterations. Fig. 2(a) shows the evolution of parameter θ_1 in the ten chains to the target posterior distribution (equation (4)), with different chains plotted in different colors. The figure indicates that the MCMC simulation in UCODE_2014 can successfully jump from one mode to the other with visits in both modes during chain evolution. This can be clearly seen in Fig. 2(b) where the evolutions of two chains are plotted by connecting successive parameters within a given chain.

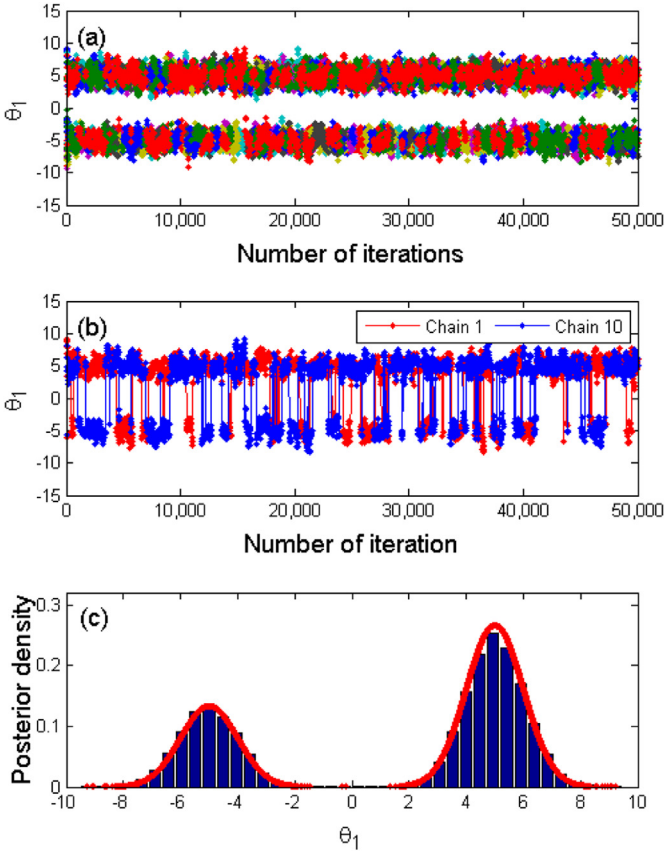

Fig. 2. Results for the 10-dimensional bimodal mathematical function. Evolution of sampled θ_1 values (a) in ten chains and (b) in two chains, with different chains represented in different colors; (c) approximated (histogram) and theoretical (red curve) marginal posterior distributions of θ_1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The mode with larger density has more visits and this is true for all the chains, resulting in a good mix of the individual paths and consequently a relatively fast convergence after about 4000 iterations. The chain convergence is diagnosed by the Gelman-Rubin $\widehat{R}$ statistic. As shown in Fig. 3(a), after about 4000 iterations the $\widehat{R}$ statistic is smaller than the threshold of 1.2, and this is similar to the result of the original MATLAB version of DREAM code as shown in Fig. 3(b). The marginal posterior distribution of the parameter θ_1 is estimated by a histogram constructed by the last 50% of the samples in the ten chains (total $0.5 \times 50{,}000 \times 10 = 250{,}000$ samples), as exhibited in Fig. 2(c). The sampled histogram demonstrates good agreement with the theoretical distribution. This simple example demonstrates that the MCMC capability of UCODE_2014 correctly simulates the underlying multimodal posterior distribution. Fig. 3 and comparison of Fig. 2 with Fig. 2 of Vrugt et al. (2009) qualitatively indicate that the DREAM algorithm has been correctly programmed and integrated into UCODE_2014.

4.2. Case study II: a 100-dimensional multivariate Gaussian function

The second case study tests how the DREAM implementation copes with a high dimensional problem by simulating a known posterior probability density function with 100 dimensions. A 100-dimensional (100 parameters) multivariate Gaussian function with means of zeros was adapted from Vrugt et al. (2009). The covariance matrix was set with the ith diagonal term equal i, and all pairwise correlations were 0.5, i.e., the covariance between the first and the last parameters is 5.0. As in Vrugt et al. (2009), the initial sample is drawn from two uniform distributions, $\mathbf{X}$~U[−5.0,15.0] and $\mathbf{X}$~U[9.9,10.0], to test the performance of our program for both under- and over-dispersed initial distributions.

One hundred chains were launched in each initial distribution and each chain evolved 10,000 iterations resulting in 1,000,000 function evaluations. Fig. 4 illustrates the evaluation of the sampled mean of x_1, the sampled standard deviations of x_1 and x_{100}, and the sampled covariance between x_1 and x_{100} for the two initial distributions. These four posterior moments were calculated using the 50% of the most recent samples. For example, when the number of function evaluations is 1,000,000, the moments were calculated by the last 50% of the samples in all the 100 chains. The true values of the four moments are 0.0, 1.0, 10.0, and 5.0 respectively, and they are separately indicated in the figure with different colored symbols. The figure indicates that the simulation smoothly approaches the true values in about 300,000 function evaluations for both initial distributions. In fact, in both cases, the chains converge after about 5000 iterations as indicated in Fig. 5. Fig. 5 plots the Gelman-Rubin $\widehat{R}$ statistics of all 100 parameters and the $\widehat{R}$ values are smaller than 1.2 after about 5000 iterations. The accuracy of our program is evaluated by the average normalized Euclidean distance (D) to the true mean and the true standard deviation as defined in the equation (6) of Vrugt et al. (2009). The sampled mean and standard deviation were calculated with the last 50% of the samples in all 100 chains, i.e., the samples after chain convergence. The D values of the two initial distributions are 0.042 and 0.034, respectively. This suggests that the MCMC capability of UCODE_2014 can efficiently and effectively simulate the 100-dimensional posterior distribution.

4.3. Case study III: a groundwater reactive transport model

In the third case study, the MCMC capability of UCODE_2014 is used to approximate the posterior distributions of model parameters and predictions and to evaluate the parametric and predictive uncertainties. To provide a test, a synthetic problem is considered.

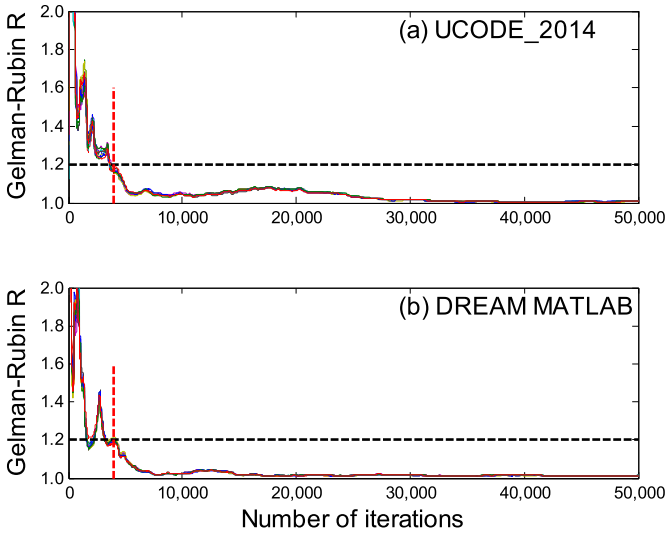

Fig. 3. Plots of Gelman-Rubin $\widehat{R}$ statistics of all the ten parameters (different parameter with different colors) based on (a) UCODE_2014 and (b) DREAM MATLAB codes. The threshold of $\widehat{R}$ value 1.2 is indicated by the black dash lines. The red dashed vertical lines indicate that after about 4000 iterations the $\widehat{R}$ values are smaller than 1.2, suggesting chain convergence. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

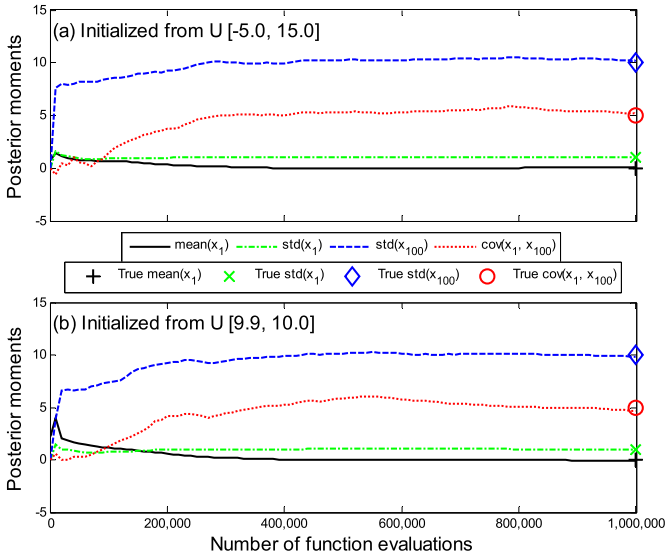

Fig. 4. Evalution of the sampled mean of x_1, the sampled standard deviations (std) of x_1 and x_{100}, and the sampled covariance (cov) between x_1 and x_{100} for the 100-dimensional multivariate Gaussian function initialized from uniform distribution (a) U [−5.0, 15.0] and (b) U [9.9, 10.0]. The true values of the four entities are separately indicated with different colored symbols at the right hand side in the figures.

The synthetic problem is adapted from Kohler et al. (1996). Kohler et al. (1996) conducted eight column experiments to study uranium reactive transport and used seven alternative surface complex models (C1−C7) to simulate the uranium adsorption. In their study, the seven models were calibrated against concentration data from their experiments 1, 2, and 8.

In the synthetic study, model C4 of Kohler et al. (1996) is considered. As shown in Table 1, the model has two surface functional groups: the weak site (S_1OH) and the strong site (S_2OH). There are four parameters all expressed in their logarithm forms: three surface complexation formation constants (LogK1, LogK2, and LogK3) and a fraction of site density for the strong site (LogSite).

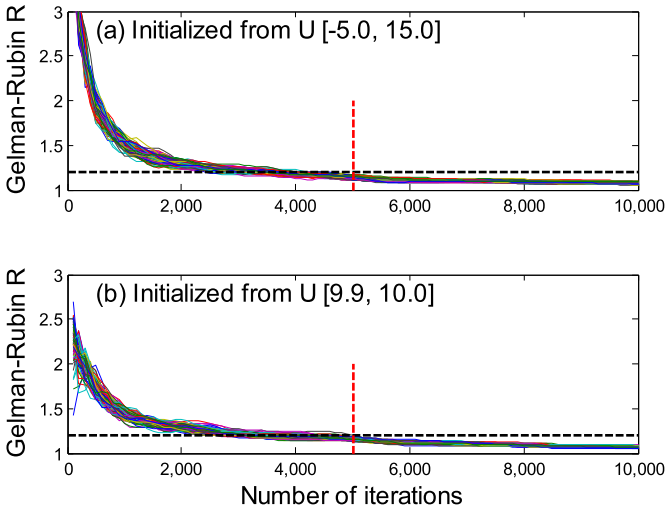

Fig. 5. Plots of Gelman-Rubin $\widehat{R}$ statistics of all 100 parameters (different parameter with different colors) for the 100-dimensional Gaussian function initialized from uniform distribution (a) U [-5.0, 15.0] and (b) U [9.9, 10.0]. The threshold of 1.2 is indicated by the black dash lines. The red dashed vertical lines indicate that after about 5000 iterations the $\widehat{R}$ values are smaller than 1.2, suggesting chain convergence. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

These four parameters are treated as random variables, as their values cannot be determined with certainty. Synthetic concentration data are first generated by simulating the model with the parameter values listed in Table 1 under the chemical conditions of experiments 1, 2, and 8 of Kohler et al. (1996). The computer code RATEQ developed by Curtis (2005) was used to generate 120 uranium concentrations. The concentrations are corrupted with measurement errors that follow a multi-Gaussian distribution with zero mean and diagonal covariance matrix with the coefficient of variation of 3%. The 120 noisy data were then used to define the objective function used in the model calibration for regression uncertainty evaluation and the likelihood function (equation (2)) used in the MCMC simulations. The uncertainty analysis is also conducted for the uranium concentrations simulated at 118 time steps under the chemical conditions of experiment 4 of Kohler et al. (1996). It should be noted that only parametric uncertainty is considered in evaluation of the predictive uncertainties in this study. The work of Lu et al. (2013) and Shi et al. (2014) showed that for the similar problem model uncertainty dominates over parametric uncertainty.

We first use UCODE_2014's optimization capability and measures of model nonlinearity to explore the model's complexity.

Table 1
Surface complexation reactions and true parameter values of the reactive transport model. Total site density used in the model is 1.3×10^{-3} mol/L_{bulk}.

U(VI) surface reaction	Formation constant	Site fraction
$S_1OH + UO_2^{2+} + H_2O = S_1OUO_2OH + 2H^+$	Log(K1) = −4.914	1.0 − Site
$S_2OH + UO_2^{2+} + H_2O = S_2OUO_2OH + 2H^+$	Log(K2) = −3.3568	Log(Site) = −1.7104
$S_2OH + UO_2^{2+} = S_2OUO_2^+ + H^+$	Log(K3) = 0.9690	

Then we use UCODE_2014 and a fast-run post processing program of LINEAR_UNCERTAINTY to calculate linear confidence intervals of the four parameters based on equation (5) and of the 118 concentrations of experiment 4 based on equation (6).

$$\widehat{\boldsymbol{\theta}} \pm 2\sqrt{\mathrm{diag}(\mathrm{Cov}_{\widehat{\boldsymbol{\theta}}})}, \quad \text{where } \mathrm{Cov}_{\widehat{\boldsymbol{\theta}}} = s^2\left(\mathbf{X}^T\mathbf{C}^{-1}\mathbf{X}\right)^{-1} \tag{5}$$

$$\widehat{\mathbf{y}} \pm 2\sqrt{\mathrm{diag}\left(\mathrm{Cov}_{\widehat{\mathbf{y}}}\right)}, \quad \text{where } \mathrm{Cov}_{\widehat{\mathbf{y}}} = s^2\mathbf{Z}\left(\mathbf{X}^T\mathbf{C}^{-1}\mathbf{X}\right)^{-1}\mathbf{Z}^T \tag{6}$$

where $\widehat{\boldsymbol{\theta}}$ is the parameter estimates, $\widehat{\mathbf{y}}$ is the prediction vector, s^2 is the estimated error variance, $\mathbf{C}$ is the covariance matrix of errors used in equation (2), $\mathbf{X}$ is the sensitivity matrix of the observations with respect to parameters evaluated at $\widehat{\boldsymbol{\theta}}$, $\mathbf{Z}$ is the corresponding sensitivity matrix of the predictions, and symbol diag means the diagonal terms of the covariance matrix, i.e., the variance.

Several parameter optimization runs with different starting values suggest absence of any local minima for this relatively simple reactive transport problem because all these runs end up with the same optimal parameter estimates. The relatively small value, 0.21, of Beales nonlinearity measure indicates that the model is not very nonlinear around the optimization identified minimum because the value of 0.21 is about half of the nonlinear criterion of 0.41. Fig. 6 plots the 95% linear confidence intervals of the four parameters and Fig. 7 plots the intervals of the 118 predictions. Both figures indicate that the intervals include the true values. The presence of the one global minimum and the small model nonlinearity suggest that the regression confidence intervals and Bayesian credible intervals may be close to each other, according to Lu et al. (2012).

The main input file for the MCMC analysis is shown in Fig. 8. Keyword MCMC in the UCODE_Control_Data input block launches the MCMC simulation. The MCMC_Controls input block provides information related to the simulation, e.g., the number of chains, the maximum number of generated parameter samples for each chain and the way to start the chains. For most of these settings, default values are used if the keyword does not appear. The MCMC_Prior_PDF input block replaces the Parameter_Data and Linear_Prior_Information input blocks used in regression to define the parameters and supplies the parameter prior information. The MCMC_Prior_PDF input block provides the parameter names, the physically reasonable parameter ranges, the prior distribution type and its associated statistics. These are defined to start the chains and calculate the prior density. For each parameter, the parameter range is defined by keywords MCLowerConstraint and MCUpperConstraint as required to ensure a meaningful forward run. The default prior distribution is uniform bounded by the given parameter range, so for a problem with uniform prior only three keywords are required as shown in Fig. 8. The Observation_Data input block remains unchanged from other UCODE_2014 modes and is used to evaluate the likelihood function. In Fig. 8 this and other input blocks use information from separate files that are not shown; e.g., the file Expt128.obs used in Observation_Data input block. For this problem the observation weights are calculated using a coefficient of variation of 3% and the measured concentrations all exceed zero.

The generated parameter sample is linked to the forward model run through input blocks under headings c and f in Fig. 8. A parallel run can be launched by setting the keyword Parallel as yes in the Parallel_Control input block with the processors information provided in the Parallel_Runners input block.

In this case study, with sufficient processors available ten chains are run in parallel for efficiency. Each chain evolves 10,000 iterations by drawing the first parameter sample from a uniform

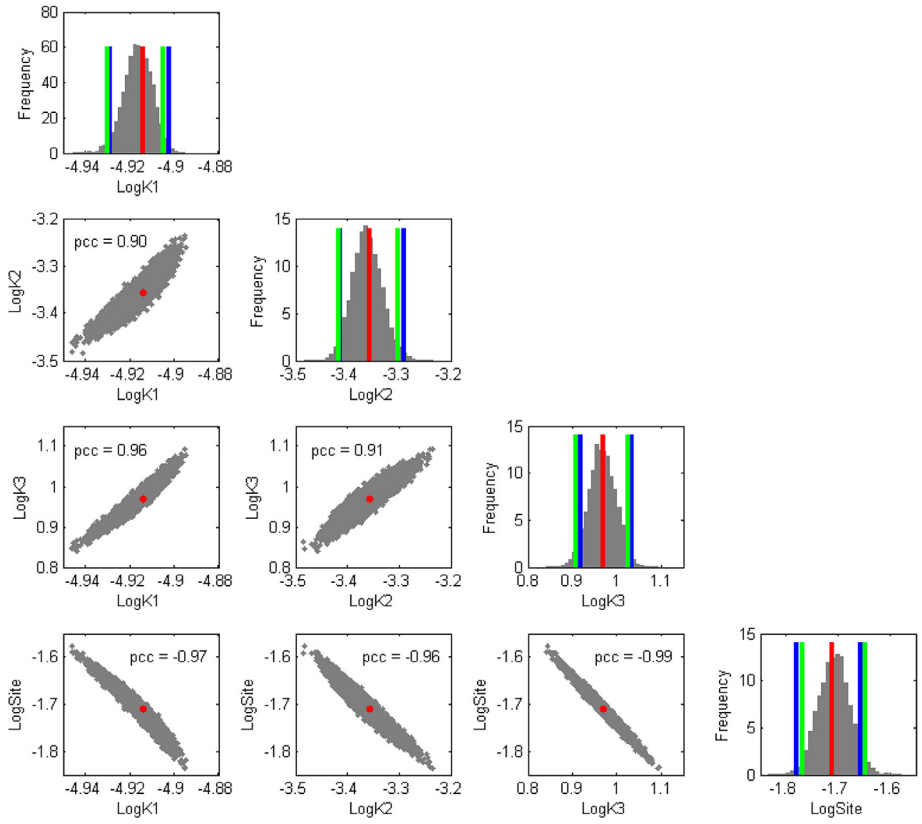

Fig. 6. The simulated marginal posterior distributions of the four parameters in the reactive transport model. The true parameter values are indicated in red, the 95% confidence intervals (9 model runs) in blue and 95% credible intervals (100,000 model runs) in green. The parameter correlation coefficients (pcc) between two parameters are shown in the off diagonal graphs to indicate their correlation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

distribution with the ranges shown in Fig. 8. Ten processors are used though for brevity Fig. 8 only lists two. The maximum of 10,000 samples generated in one chain (i.e., 100,000 samples in ten chains) is deemed sufficient to explore the target posterior distribution and to estimate the parametric uncertainty for this relatively simple problem with four parameters and one mode. For some complex problems with a large number of parameters and/or multiple modes, more samples may be required. If needed, parameter sampling can be continued by restarting the MCMC run.

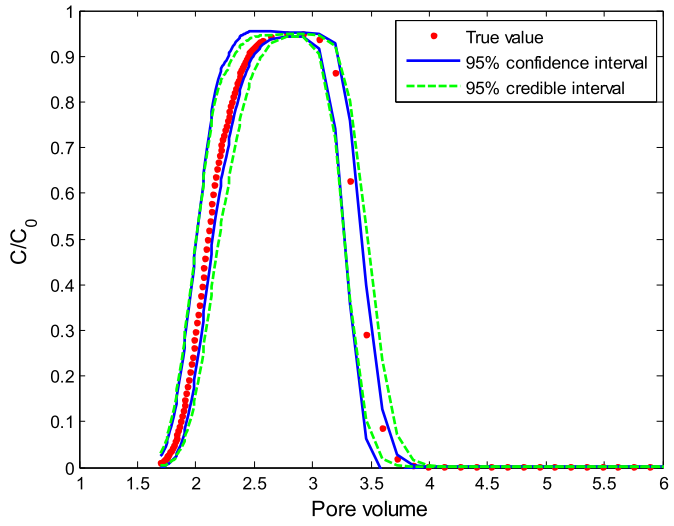

Fig. 7. The true predicted values of experiment 4 (red points) and the 95% confidence intervals (blue lines) and credible intervals (green lines) for the reactive transport model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As an example, the MCMC evolution of LogK1 and its sample mean for the ten chains are shown in Fig. 9(a) and (b), respectively, with each chain plotted by a specific color. Both figures indicate that the chains converge around iteration 2,000, from which chains mix well and the sample mean steadily approaches a constant value. The convergence around iteration 2000 is also suggested by the Gelman-Rubin $\widehat{R}$ statistic which is continuously less than 1.2 after iteration 2000. In addition, Fig. 9(a) indicates that the marginal posterior distribution of LogK1 has only one mode, and the MCMC capability of UCODE_2014 can successfully generate samples around the mode. The true value of LogK1 is plotted in Fig. 9(b) as a red dot. The figure illustrates that the chains smoothly and quickly converge to the target value. If it is known that the parameter posterior distribution has only one mode, then the convergence rate can be further improved by initializing the chains from the parameter optimization results. The information about modes can be inferred from multiple optimization runs such as those conducted in this study. For this test problem without local minima and with only one mode, the MCMC simulation starting from the optimization results converges around iteration 1,000, accelerating the efficiency by about a factor of two. Though the results are not shown, the other three parameters have very similar performance as depicted in Fig. 9 of LogK1.

The last 50% of the samples in the ten chains (i.e., $0.5 \times 100{,}000 = 50{,}000$ samples) are used to construct histograms and estimate the marginal posterior distribution of each individual

```
#-----------------------------------------------------
# a. UCODE_CONTROL INFORMATION
#-----------------------------------------------------
BEGIN UCODE_CONTROL_DATA KEYWORDS
  MCMC=yes
END UCODE_CONTROL_DATA
#-----------------------------------------------------
# b. MCMC_CONTROLS INPUT BLOCK
#-----------------------------------------------------
BEGIN MCMC_CONTROLS KEYWORDS
  Nchain=10    MaxSamples=10000
  Npair=3    NCR=3    JumpStep=5
  GelmanR=1.2    PrintStep=10
  UseRegResult=no    Restart=no
END MCMC_CONTROLS
#-----------------------------------------------------
# c. COMMAND FOR PROCESS MODEL
#-----------------------------------------------------
BEGIN MODEL_COMMAND_LINES KEYWORDS
  Command='./Run.bat'    CommandId='Rateq'
END MODEL_COMMAND_LINES
#-----------------------------------------------------
# d. MCMC_PRIOR_PDF INPUT BLOCK
#-----------------------------------------------------
BEGIN MCMC_Prior_PDF TABLE
  NRow=4    NCol=3    COLUMNLABELS
  ParamName    MCLowerConstraint MCUpperConstraint
  LogK1              -5.9d0              -4.0d0
  LogK2              -4.4d0              -2.3d0
  LogK3               0.1d0               2.0d0
  LogSite            -2.7d0              -0.7d0
END MCMC_Prior_PDF
#-----------------------------------------------------
# e. OBSERVATION_DATA INPUT BLOCK
#-----------------------------------------------------
BEGIN OBSERVATION_DATA FILES
  Expt128.obs
END OBSERVATION_DATA
#-----------------------------------------------------
# f. PROCESS MODEL INFORMATION
#-----------------------------------------------------
BEGIN MODEL_INPUT_FILES KEYWORDS
  ModInFile=par.in    TemplateFile=par.tpl
END MODEL_INPUT_FILES

BEGIN MODEL_OUTPUT_FILES KEYWORDS
  ModOutFile=sim.out    InstructionFile=sim.inst    Category=obs
END MODEL_OUTPUT_FILES
#---------------------------------
# g. PARALLEL PROCESSING
#---------------------------------
BEGIN PARALLEL_CONTROL
  Parallel=yes    OperatingSystem=Linux
END PARALLEL_CONTROL

BEGIN PARALLEL_RUNNERS TABLE
  NRow=2    NCol=3    COLUMNLABELS
  RunnerName    RunnerDir    RunTime
  Runner1          ../run1/        1000
  Runner2          ../run2/        1000
END PARALLEL_RUNNERS
```

Fig. 8. The main input file of UCODE_2014 for an MCMC run. Inputs unique to MCMC are highlighted in gray. For this problem the MCMC_Prior_PDF input block defines only uniform prior PDFs; eleven types of distributions are supported.

parameter as depicted in the graphs along the diagonal of Fig. 6. The graphs indicate that the sampled marginal posterior distributions of the four parameters have only one mode which is centered on the true values, and they are Gaussian-like with negligible skewness. These parameter samples are also used to estimate the marginal posterior distributions of any two parameters as depicted in the lower-triangle area of Fig. 6. These graphs illustrate the correlation between two parameters and their correlation coefficients are indicated in the graphs. For example, the graph in the bottom-left corner suggests that parameters LogK1 and LogSite are strongly

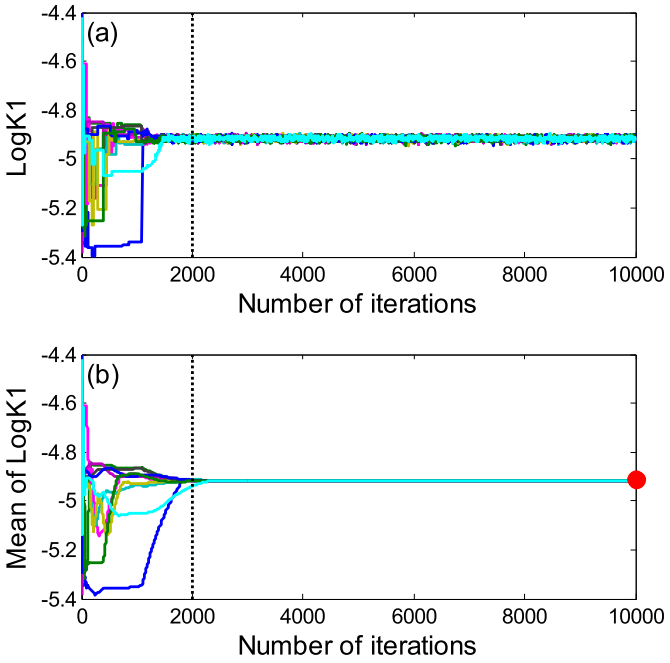

Fig. 9. The evolution of (a) parameter LogK1 and (b) its sample mean for ten chains in the reactive transport model. The vertical black dashed lines define the end of the burn-in period, which indicate that at iteration 2,000, the chains are converged. The true value of the parameter is represented by a red dot at the right hand side of (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

negatively correlated and their correlation coefficient is −0.97. This is physically reasonable, because, if the fraction of strong site increases (i.e., LogSite increases), then the affinity of uranium for the weak site (controlled by logK1) needs to decrease so that the simulated uranium concentrations can match the observed concentrations. Therefore, the MCMC simulation confirms our understanding of the uranium reactive processes and may help advance our understanding of the processes.

The parametric uncertainties of the four parameters are evaluated by their 95% credible intervals using the equal-tailed method. First, for each parameter the 50,000 samples are ordered from smallest value to largest. Then, the lower and upper limits of the 95% credible interval equal the values that fall exactly at the 2.5% and 97.5% levels, as shown by the green lines in Fig. 6. The fact that the two green lines enclose the red lines (representing the true parameter values) in all the histograms demonstrates that the intervals are capable of including all the true parameter values. In Fig. 6, the Bayesian credible intervals are close to the regression confidence intervals for this test problem without local minima and with small model nonlinearity. Similar good agreement was reported recently for a groundwater flow problem (Lu et al., 2012).

For each of the 50,000 converged parameter samples, the predictions are evaluated and the predictive uncertainties are quantified in the same manner as for the parametric uncertainties by using the MCMC outputs of a prediction run. The main input file used to generate parameter samples (Fig. 8) can be used to evaluate prediction samples with little alteration. The prediction run is launched by adding the keyword MCMC_PREDICTION to the UCODE_Control_Data input block and setting it to yes. The keyword ItStartPred needs to be added to MCMC_Controls input block to specify the iteration number after which the parameter samples are used to evaluate predictions (e.g., ItStartPred is $0.5 \times 10{,}000 = 5000$ in this study). ItStartPred should not be smaller than the iteration where chains converge as diagnosed by the Gelman-Rubin $\hat{R}$ to

prevent parameters from the burn-in period from being used for evaluating predictions. ItStartPred is usually equal to half of the maximum samples generated in one chain, and is the default value set in UCODE_2014. The MCMC_Prior_PDF input block can remain unchanged, although only the keyword ParamName is used. The Observation_Data input block is ignored and a Prediction_Data input block needs to be added. The input blocks under headings c and f in Fig. 8 related to the forward model run need to be changed to those corresponding to the prediction model run. The parallel processing input block can be maintained to conduct the needed simulations on multiple processors.

For each prediction of experiment 4, the 50,000 prediction samples are used to calculate its 95% credible interval using the equal-tailed method. The results are plotted in Fig. 7 together with the confidence intervals. The figure indicates that the credible intervals enclose all the true values, as expected for this simple problem. The Bayesian credible intervals and the regression confidence intervals are numerically close to each other. The largest differences occur just before the peak and at the right tail.

The MCMC simulation of this model took approximately 17 days to complete on a Linux system with 10 processors running in parallel. Almost all the CPU time is consumed by repeatedly running the forward model (each forward model run took about 1 min which is relatively small for a reactive transport model); a negligible portion of the CPU time is used by the sampling algorithm. If the same MCMC simulation is conducted with one processor, the CPU time is about 70 days. This large amount of time can be significantly reduced by surrogate modeling like the work conducted in Zhang et al. (2013), which applied the MCMC simulation to the surrogate model. Currently the surrogate modeling approach is not incorporated in UCODE_2014.

5. Concluding remarks

Regression and Bayesian uncertainty analysis are two different ways to quantify parametric and predictive uncertainties. While the regression confidence intervals are computationally frugal and easy to compute, the computationally expensive Bayesian credible intervals are more accurate for highly nonlinear problems with local minima. The ability of UCODE_2014 to calculate both kinds of intervals provides great flexibility in evaluating parametric and predictive uncertainties.

This work integrates the advanced MCMC algorithm, DREAM, into UCODE_2014 for Bayesian uncertainty analysis. This enables UCODE_2014 to efficiently handle high-dimensional and multimodal problems while taking advantage of UCODE structures to obtain straightforward and flexible execution. With the template and instruction files of UCODE, the uncertainty analysis can be assessed for any process models with ASCII-based inputs and outputs without modifying source code. With the parallel computing capability based on UCODE's robust dispatcher-runner protocol, the Bayesian uncertainty analysis can be evaluated with relatively high efficiency. In addition, UCODE_2014 provides versatile ways to initialize the MCMC process and a proper selection can accelerate the chain convergence and save computational time. It also provides a variety of parameter prior distributions which makes its application more flexible. Most importantly, the program can be used for not only the parametric but also the predictive uncertainty analysis with little alteration of the input files.

This work uses a groundwater reactive transport model to illustrate that UCODE_2014 can be used for Bayesian uncertainty analysis of a variety of environmental problems. When applied to large-scale inverse problems, e.g., highly heterogeneous problems, MCMC methods face two major challenges. First, the complex process models may make MCMC simulation computationally unaffordable to evaluate posterior probability density for any parameter sample. In addition, the high-dimensional parameter spaces make the exploration of the posterior parameter distribution difficult sometimes even prohibitive. To resolve the first challenge, currently there are two major strategies: (1) develop advanced MCMC algorithms to improve computationally efficiency by reducing the needed number of process model executions, i.e., MT-DREAMzs (Laloy and Vrugt, 2012); and (2) apply model surrogate methods to improve computationally efficiency by evaluating the computationally frugal surrogate model instead of the actual process model, i.e., Zhang et al. (2013) and Laloy et al. (2013). The second challenge is more difficult because the high dimension usually means large model nonlinearity and many local minima. For those problems, currently none of the MCMC methods can guarantee that all local minima will be found, and surrogate methods are no longer absolutely computationally competitive because building the surrogate model can require a large number of process model runs. Combining the advanced MCMC algorithm with some dimensionality reduction techniques may be a good solution.

Acknowledgment

This work was supported in part by NSF-EAR grant 0911074, DOE-SBR grant DE-SC0002687, DOE Early Career Award, DE-SC0008272, and National Natural Science Foundation of China grants, 51328902. The authors thank John Doherty and Jasper Vrugt for providing the MICA and DREAM codes.

References

Adams, B.M., Ebeida, M.S., Eldred, M.S., Jakeman, J.D., Swiler, L.P., 2013. Dakota, A Multilevel Parallel Object-oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Sandia National Laboratories, SAND2010–2183, 329 pp.

Anderman, E.R., Hill, M.C., 1999. A new multi-stage groundwater transport inverse method, Presentation, evaluation, and implications. Water Resour. Res. 35 (4), 1053–1063.

Banta, E.R., Poeter, E.P., Doherty, J.E., Hill, M.C., 2006. JUPITER: joint universal parameter identification and evaluation of reliability – an application programming interface (API) for model analysis. U.S. Geol. Surv. Tech. Methods, 06–E1, 268 pp.

Banta, E.R., Hill, M.C., Poeter, E.P., Doherty, J.E., Babendreier, J., 2008. Building model analysis applications with the Joint Universal Parameter IdenTification and Evaluation of Reliability (JUPITER) API. Comput. Geosci. 34, 310–319.

Bates, D.M., Watts, D.G., 1988. Nonlinear Regression Analysis and its Applications. John Wiley and Sons, New York, p. 365.

Box, E.P., Tiao, G.C., 1992. Bayesian Inference in Statistical Analysis. Wiley, New York, 588 pp.

Curtis, G.P., 2005. Documentation and Applications of the Reactive Geochemical Transport Model RATEQ. U. S. Nuclear Regulatory Commission NUREG report, 97 pp.

Doherty, J., 2003. MICA: Model-independent Markov Chain Monte Carlo Analysis. Watermark Numerical Computing, Brisbane, Australia.

Doherty, J., 2005. PEST: Software for Model-Independent Parameter Estimation. Watermark Numerical computing, Australia. Available from: http://www.sspa.com/pest.

Finsterle, S., Zhang, Y., 2011a. Error handling strategies in multiphase inverse modeling. Comput. Geosci. 37, 724–730. http://dx.doi.org/10.1016/j.cageo.2010.11.009.

Finsterle, S., Zhang, Y., 2011b. Solving iTOUGH2 simulation and optimization problems using the PEST protocol. Environ Model. Softw. 26, 959–968.

Gallagher, M., Doherty, J., 2007. Parameter estimation and uncertainty analysis for a watershed model. Environ. Model. Softw. 22, 1000–1020.

Gelman, A., Rubin, D.B., 1992. Inference from iterative simulation using multiple sequences. Stat. Sci. 7 (4), 457–472.

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 1995. Bayesian Data Analysis. Chapman and Hall, London, p. 696.

Haario, H., Laine, M., Mira, A., Saksman, E., 2006. DRAM: efficient adaptive MCMC. Stat. Comp. 16, 339–354.

Hill, M.C., Tiedeman, C.R., 2007. Effective Calibration of Ground Water Models, with Analysis of Data, Sensitivities, Predictions, and Uncertainty. John Wiley & Sons, New York, p. 480.

Jaynes, E.T., 1976. Confidence intervals vs Bayesian intervals. In: Harper, W.L., Hooker, C.A. (Eds.), Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science. D. Reidel Publ, Dordrecht, Netherlands, pp. 175–257.

Joseph, J.F., Guillaume, J.H.A., 2013. Using a parallelized MCMC algorithm in R to identify appropriate likelihood functions for SWAT. Environ. Model. Softw. 46, 292–298.

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

Kohler, M., Curtis, G.P., Kent, D.B., Davis, J.A., 1996. Experimental investigation and modeling of uranium (VI) transport under variable chemical conditions. Water Resour. Res. 32 (12), 3539–3551.

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

Laloy, E., Rogiers, B., Vrugt, J.A., Mallants, D., Jacques, D., 2013. Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov Chain Monte Carlo simulation and polynomial chaos expansion. Water Resour. Res. 49, 2664–2682.

Liu, X., Cardiff, M.A., Kitanidis, P.K., 2010. Parameter estimation in nonlinear environmental problems. Stoch. Environ. Res. Risk Assess. 24, 1003–1022. http://dx.doi.org/10.1007/s00477-010-0395-y.

Lu, D., Ye, M., Hill, M.C., 2012. Analysis of regression confidence intervals and Bayesian credible intervals for uncertainty quantification. Water Resour. Res. 48, W09521. http://dx.doi.org/10.1029/2011WR011289.

Lu, D., Ye, M., Meyer, P.D., Curtis, G.P., Shi, X., Niu, X.-F., Yabusaki, S.B., 2013. Effects of error covariance structure on estimation of model averaging weights and predictive performance. Water Resour. Res. 49 http://dx.doi.org/10.1002/wrcr.20441.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equations of state calculations by fast computing machines. J. Chem. Phys. 21, 1087–1091.

Poeter, E.P., Hill, M.C., Banta, E.R., Mehl, S.W., Christensen, S., 2005. UCODE_2005 and six other computer codes for universal sensitivity analysis, inverse modeling, and uncertainty evaluation. U.S. Geol. Surv. Tech. Methods 6–A11, 283.

Poeter, E.P., Hill, M.C., Lu, D., Mehl, S.W., 2014. UCODE_2014, with New Capabilities to Define Parameters Unique to Predictions, Calculate Weights Using Simulated Values, Estimate Parameters with SVD, and Evaluate Uncertainty with MCMC. International Ground Water Modeling Center Report, to appear.

Robert, C., Casella, G., 2004. Monte Carlo Statistical Method, second ed. Springer. 645 pp.

Shi, X., Ye, M., Finsterle, S., Wu, J., 2012. Comparing nonlinear regression and Markov chain Monte Carlo methods for assessment of predictive uncertainty in vadose zone modeling. Vadose Zone J. 11 (4) http://dx.doi.org/10.2136/vzj2011.0147.

Shi, X., Ye, M., Curtis, G.P., Miller, G.L., Meyer, P.D., Kohler, M., Yabusaki, S., Wu, J., 2014. Assessment of parametric uncertainty for groundwater reactive transport modeling. Water Resour. Res. 50 http://dx.doi.org/10.1002/2013WR013755.

Smith, T., Sharma, A., Marshall, L., Mehrotra, R., Sisson, S., 2010. Development of a formal likelihood function for improved Bayesian inference of ephemeral catchments. Water Resour. Res. 46, W12551. http://dx.doi.org/10.1029/2010WR009514.

Ter Braak, C.J.F., 2006. A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces. Stat. Comp. 16, 239–249.

Vrugt, J.A., Bouten, W., 2002. Validity of first-order approximations to describe parameter uncertainty in soil hydraulic models. Soil. Sci. Soc. Am. J. 66, 1740–1751. http://dx.doi.org/10.2136/sssaj2002.1740.

Vrugt, J.A., Nualláin, B.Ó., Robinson, B.A., Bouten, W., Dekker, S.C., Sloot, P.M.A., 2006. Application of parallel computing to stochastic parameter estimation in environmental models. Comput. Geosci. 32, 1139–1155.

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

Vrugt, J.A., ter Braak, C.J.F., Diks, C.G.H., Robinson, B.A., Hyman, J.M., Higdon, D., 2009. Accelerating Markov Chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. Int. J. Nonlinear Sci. Numer. Simul. 10 (3), 271–288.

Zhang, G., Lu, D., Ye, M., Gunzburger, M., Webster, C., 2013. An adaptive sparse-grid high-order stochastic collocation method for Bayesian inference in groundwater reactive transport modeling. Water Resour. Res. 49 http://dx.doi.org/10.1002/wrcr.20467.