

## DREAM Suite: Technical manual

Jasper A. Vrugt<sup>a,b,d</sup>, and Mirek Šejna<sup>c</sup><sup>a</sup>*Department of Civil and Environmental Engineering, University of California Irvine, 4130 Engineering Gateway, Irvine, CA 92697-2175*<sup>b</sup>*Department of Earth Systems Science, University of California Irvine, Irvine, CA.*<sup>c</sup>*PC-Progress s.r.o., Koruní 2569/108a, Prague, 101 00, Czech Republic*<sup>d</sup>*Note: please cite this manuscript as Vrugt (2016), Environmental Modeling & Software, vol. 75, pp. 273-316, 10.1016/j.envsoft.2015.08.013, 2016.*

---

**Abstract**

Bayesian inference has found widespread application and use in science and engineering to reconcile Earth system models with data, including prediction in space (interpolation), prediction in time (forecasting), assimilation of observations and deterministic/stochastic model output, and inference of the model parameters. Bayes theorem states that the posterior probability,  $p(H|\tilde{\mathbf{Y}})$  of a hypothesis,  $H$  is proportional to the product of the prior probability,  $p(H)$  of this hypothesis and the likelihood,  $L(H|\tilde{\mathbf{Y}})$  of the same hypothesis given the new observations,  $\tilde{\mathbf{Y}}$ , or  $p(H|\tilde{\mathbf{Y}}) \propto p(H)L(H|\tilde{\mathbf{Y}})$ . In science and engineering,  $H$  often constitutes some numerical model,  $\mathcal{F}(\mathbf{x}, \cdot)$  which summarizes, in algebraic and differential equations, state variables and fluxes, all knowledge of the system of interest, and the unknown parameter values,  $\mathbf{x}$  are subject to inference using the data  $\tilde{\mathbf{Y}}$ . Unfortunately, for complex system models the posterior distribution is often high dimensional and analytically intractable, and sampling methods are required to approximate the target. In this manual I review the building blocks of the DiffeRential Evolution Adaptive Metropolis (DREAM) algorithm developed by Vrugt *et al.* (2008a, 2009a), and discuss its implementation within DREAM Suite. This program provides scientists and engineers with an arsenal of options and utilities to solve posterior sampling problems involving (among others) bimodality, high-dimensionality, summary statistics, bounded parameter spaces, dynamic simulation models, formal/informal likelihood functions (GLUE), diagnostic model evaluation, data assimilation, Bayesian model averaging, distributed computation, and informative/noninformative prior distributions. DREAM Suite supports parallel computing and includes convergence analysis of the sampled chain trajectories and post-processing of the results.

**Keywords:** Bayesian inference, Markov chain Monte Carlo (MCMC) simulation, Random walk Metropolis (RWM), Adaptive Metropolis (AM), Differential evolution Markov Chain (DE-MC), Prior distribution, Likelihood function, Posterior distribution, Approximate Bayesian computation (ABC), Diagnostic model evaluation, Residual analysis, Environmental modeling, Bayesian model averaging (BMA), Generalized likelihood uncertainty estimation (GLUE), Multi-processor computing

---



---

*Email address:* [jasper@uci.edu](mailto:jasper@uci.edu) (Jasper A. Vrugt)*URL:* <http://faculty.sites.uci.edu/jasper> (Jasper A. Vrugt),<http://scholar.google.com/citations?user=zKNXecUAAAAJ&hl=en> (Jasper A. Vrugt)*Preprint submitted to Manual**March 9, 2016*

## DISCLAIMER

This report documents version 1 of DREAM Suite, a software package for Bayesian inference using Markov chain Monte Carlo simulation with DREAM. The software has been verified against a large number of test cases. However, no warranty is given that the program is completely error-free. If you do encounter problems with the code, find errors, or have suggestions for improvement, please contact

Jasper A. Vrugt  
Department of Civil and Environmental Engineering  
University of California Irvine  
Irvine, CA 92617  
USA  
Phone: (505)-231-2698  
Email: [jasper@uci.edu](mailto:jasper@uci.edu)

## 1. INTRODUCTION AND SCOPE

Continued advances in direct and indirect (e.g. geophysical, pumping test, remote sensing) measurement technologies and improvements in computational technology and process knowledge have stimulated the development of increasingly complex environmental models that use algebraic and (stochastic) ordinary (partial) differential equations (PDEs) to simulate the behavior of a myriad of highly interrelated ecological, hydrological, and biogeochemical processes at different spatial and temporal scales. These water, energy, nutrient, and vegetation processes are often non-separable, non-stationary with very complicated and highly-nonlinear spatio-temporal interactions (*Wikle and Hooten*, 2010) which gives rise to complex system behavior. This complexity poses significant measurement and modeling challenges, in particular how to adequately characterize the spatio-temporal processes of the dynamic system of interest, in the presence of (often) incomplete and insufficient observations, process knowledge and system characterization. This includes prediction in space (interpolation/extrapolation), prediction in time (forecasting), assimilation of observations and deterministic/stochastic model output, and inference of the model parameters.

The use of differential equations might be more appropriate than purely empirical relationships among variables, but does not guard against epistemic errors due to incomplete and/or inexact process knowledge. Figure 1 provides a schematic overview of most important sources of uncertainty that affect our ability to describe as closely and consistently as possible the observed system behavior. These sources of uncertainty have been discussed extensively in the literature, and much work has focused on the characterization of parameter, model output and state variable uncertainty. Explicit knowledge of each individual error source would provide strategic guidance for investments in data collection and/or model improvement. For instance, if input (forcing/boundary condition) data uncertainty dominates total simulation uncertainty, then it would not be productive to increase model complexity, but rather to prioritize data collection instead. On the contrary, it would be naive to spend a large portion of the available monetary budget on system characterization if this constitutes only a minor portion of total prediction uncertainty.

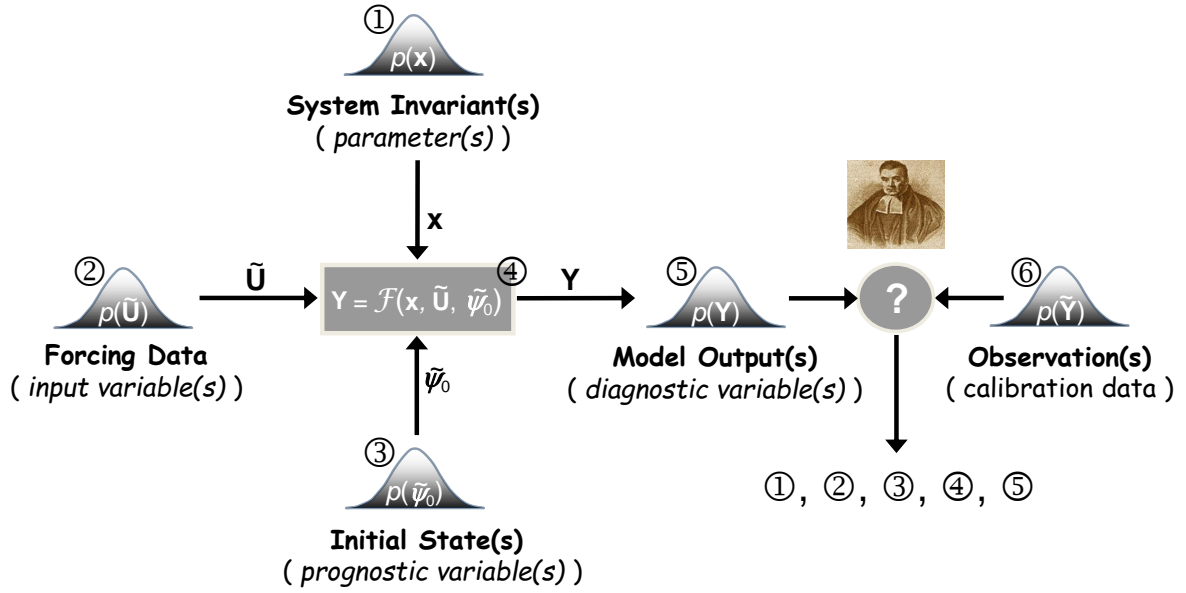


Figure 1 Schematic illustration of the most important sources of uncertainty in environmental systems modeling, including (1) parameter, (2) input data (also called forcing or boundary conditions), (3), initial state, (4) model structural, (5) output, and (6) calibration data uncertainty. The measurement data error is often conveniently assumed to be known, a rather optimistic approach in most practical situations. Question remains how to describe/infer properly all sources of uncertainty in a coherent and statistically adequate manner.

Note that model structural error (label 4) (also called epistemic error) has received relatively little attention, but is key to learning and scientific discovery (*Vrugt et al.*, 2005; *Vrugt and Sadegh*, 2013a).

The focus of this paper is on spatio-temporal models that may be discrete in time and/or space, but with processes that are continuous in both. An algorithm and program are described which can be used to derive the posterior parameter (and state) distribution, conditioned on measurements of observed system behavior. At least some level of calibration of these models is required to make sure that the simulated state variables, internal fluxes, and output variables match the observed system behavior as closely and consistently as possible. Bayesian methods have found widespread application and use to do so, in particular because of their innate ability to handle, in a consistent and coherent manner parameter, state variable, and model output (simulation) uncertainty.

If  $\tilde{\mathbf{Y}} = \{\tilde{y}_1, \dots, \tilde{y}_n\}$  signifies a discrete vector of measurements at times  $t = \{1, \dots, n\}$  which summarizes the response of some environmental system  $\mathfrak{S}$  to forcing variables  $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ . The observations or data are linked to the physical system

$$\tilde{\mathbf{Y}} \leftarrow \mathfrak{S}(\mathbf{x}^*) + \boldsymbol{\varepsilon}, \quad (1)$$

where  $\mathbf{x}^* = \{x_1^*, \dots, x_d^*\}$  are the unknown parameters, and  $\boldsymbol{\varepsilon} = \{\varepsilon_1, \dots, \varepsilon_n\}$  is a  $n$ -vector

of measurement errors. When a hypothesis, or simulator,  $\mathbf{Y} \leftarrow \mathcal{F}(\mathbf{x}^*, \tilde{\mathbf{u}}, \tilde{\boldsymbol{\psi}}_0)$  of the physical process is available, then the data can be modeled using

$$\tilde{\mathbf{Y}} \leftarrow \mathcal{F}(\mathbf{x}^*, \tilde{\mathbf{U}}, \tilde{\boldsymbol{\psi}}_0) + \mathbf{E}, \quad (2)$$

where  $\tilde{\boldsymbol{\psi}}_0 \in \boldsymbol{\Psi} \in \mathbb{R}^\tau$  signify the  $\tau$  initial states, and  $\mathbf{E} = \{e_1, \dots, e_n\}$  includes observation error (forcing and output data) as well as error due to the fact that the simulator,  $\mathcal{F}(\cdot)$  may be systematically different from reality,  $\mathfrak{S}(\mathbf{x}^*)$  for the parameters  $\mathbf{x}^*$ . The latter may arise from numerical errors (inadequate solver and discretization), and improper model formulation and/or parameterization.

By adopting a Bayesian formalism the posterior distribution of the parameters of the model can be derived by conditioning the spatio-temporal behavior of the model on measurements of the observed system response

$$p(\mathbf{x}|\tilde{\mathbf{Y}}) = \frac{p(\mathbf{x})p(\tilde{\mathbf{Y}}|\mathbf{x})}{p(\tilde{\mathbf{Y}})}, \quad (3)$$

where  $p(\mathbf{x})$  and  $p(\mathbf{x}|\tilde{\mathbf{Y}})$  signify the prior and posterior parameter distribution, respectively, and  $L(\mathbf{x}|\tilde{\mathbf{Y}}) \equiv p(\tilde{\mathbf{Y}}|\mathbf{x})$  denotes the likelihood function. The evidence,  $p(\tilde{\mathbf{Y}})$  acts as a normalization constant (scalar) so that the posterior distribution integrates to unity

$$p(\tilde{\mathbf{Y}}) = \int_{\chi} p(\mathbf{x})p(\tilde{\mathbf{Y}}|\mathbf{x})d\mathbf{x} = \int_{\chi} p(\mathbf{x}, \tilde{\mathbf{Y}})d\mathbf{x}, \quad (4)$$

over the parameter space,  $\mathbf{x} \in \chi \in \mathbb{R}^d$ . In practice,  $p(\tilde{\mathbf{Y}})$  is not required for posterior estimation as all statistical inferences about  $p(\mathbf{x}|\tilde{\mathbf{Y}})$  can be made from the unnormalized density

$$p(\mathbf{x}|\tilde{\mathbf{Y}}) \propto p(\mathbf{x})L(\mathbf{x}|\tilde{\mathbf{Y}}) \quad (5)$$

The main culprit now resides in the definition of the likelihood function,  $L(\mathbf{x}|\tilde{\mathbf{Y}})$  used to summarize the distance between the model simulations and corresponding observations. If the error residuals are assumed to be uncorrelated then the likelihood of the  $n$ -vector of error residuals can be written as follows

$$L(\mathbf{x}|\tilde{\mathbf{Y}}) = f_{\tilde{y}_1}(y_1(\mathbf{x})) \times f_{\tilde{y}_2}(y_2(\mathbf{x})) \times \dots \times f_{\tilde{y}_n}(y_n(\mathbf{x})) = \prod_{t=1}^n f_{\tilde{y}_t}(y_t(\mathbf{x})), \quad (6)$$

where  $f_a(b)$  signifies the probability density function of  $a$  evaluated at  $b$ . If we further assume

the error residuals to be normally distributed,  $e_t(\mathbf{x}) \stackrel{\mathcal{D}}{\sim} \mathcal{N}(0, \hat{\sigma}_t^2)$  then Equation (6) becomes

$$L(\mathbf{x}|\tilde{\mathbf{Y}}, \hat{\boldsymbol{\sigma}}^2) = \prod_{t=1}^n \frac{1}{\sqrt{2\pi\hat{\sigma}_t^2}} \exp \left[ -\frac{1}{2} \left( \frac{\tilde{y}_t - y_t(\mathbf{x})}{\hat{\sigma}_t} \right)^2 \right], \quad (7)$$

where  $\hat{\boldsymbol{\sigma}} = \{\hat{\sigma}_1, \dots, \hat{\sigma}_n\}$  is a  $n$ -vector with standard deviations of the measurement error of the observations. This formulation allows for homoscedastic (constant variance) and heteroscedastic measurement errors (variance dependent on magnitude of data).<sup>1</sup> For reasons of numerical stability and algebraic simplicity it is often convenient to work with the log-likelihood,  $\mathcal{L}(\mathbf{x}|\tilde{\mathbf{Y}}, \hat{\boldsymbol{\sigma}}^2)$  instead

$$\mathcal{L}(\mathbf{x}|\tilde{\mathbf{Y}}, \hat{\boldsymbol{\sigma}}^2) = -\frac{n}{2} \log(2\pi) - \sum_{t=1}^n \{\log(\hat{\sigma}_t)\} - \frac{1}{2} \sum_{t=1}^n \left( \frac{\tilde{y}_t - y_t(\mathbf{x})}{\hat{\sigma}_t} \right)^2. \quad (8)$$

If the error residuals,  $\mathbf{E}(\mathbf{x}) = \tilde{\mathbf{Y}} - \mathbf{Y}(\mathbf{x}) = \{e_1(\mathbf{x}), \dots, e_n(\mathbf{x})\}$  exhibit temporal (or spatial) correlation then one can try to take explicit account of this in the derivation of the log-likelihood function. For instance, suppose the error residuals assume an AR(1)-process

$$e_t(\mathbf{x}) = c + \phi e_{t-1}(\mathbf{x}) + \eta_t, \quad (9)$$

with  $\eta_t \stackrel{\mathcal{D}}{\sim} \mathcal{N}(0, \hat{\sigma}_t^2)$ , expectation  $\mathbb{E}[e_t(\mathbf{x})] = c/(1-\phi)$ , and variance  $\text{Var}[e_t(\mathbf{x})] = \hat{\sigma}_t^2/(1-\phi^2)$ . This then leads to the following formulation of the log-likelihood (derivation in statistics textbooks)

$$\begin{aligned} \mathcal{L}(\mathbf{x}|\tilde{\mathbf{Y}}, c, \phi, \hat{\boldsymbol{\sigma}}^2) = & -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log[\hat{\sigma}_1^2/(1-\phi^2)] - \frac{(e_1(\mathbf{x}) - [c/(1-\phi)])^2}{2\hat{\sigma}_1^2/(1-\phi^2)} \\ & - \sum_{t=2}^n \{\log(\hat{\sigma}_t)\} - \frac{1}{2} \sum_{t=2}^n \left( \frac{(e_t(\mathbf{x}) - c - \phi e_{t-1}(\mathbf{x}))}{\hat{\sigma}_t} \right)^2 \end{aligned} \quad (10)$$

where  $|\phi| < 1$  signifies the first-order autoregressive coefficient. If we assume  $c$  to be zero (absence of long-term trend) then Equation (10) reduces, after some rearrangement, to

$$\begin{aligned} \mathcal{L}(\mathbf{x}|\tilde{\mathbf{Y}}, \phi, \hat{\boldsymbol{\sigma}}^2) = & -\frac{n}{2} \log(2\pi) + \frac{1}{2} \log(1-\phi^2) - \frac{1}{2} (1-\phi^2) \hat{\sigma}_1^{-2} e_1(\mathbf{x})^2 \\ & - \sum_{t=2}^n \{\log(\hat{\sigma}_t)\} - \frac{1}{2} \sum_{t=2}^n \left( \frac{(e_t(\mathbf{x}) - \phi e_{t-1}(\mathbf{x}))}{\hat{\sigma}_t} \right)^2, \end{aligned} \quad (11)$$

and the nuisance variables  $\{\phi, \boldsymbol{\sigma}\}$  are subject to inference with the model parameters,  $\mathbf{x}$

---

<sup>1</sup>If homoscedasticity is expected and the variance of the error residuals,  $s^2 = \frac{1}{n-1} \sum_{t=1}^n (e_t(\mathbf{x}))^2$  is taken as sufficient statistic for  $\sigma^2$ , then one can show that the likelihood function simplifies to  $L(\mathbf{x}|\tilde{\mathbf{Y}}) \propto \sum_{t=1}^n |e_t(\mathbf{x})|^{-n}$

using the observed data,  $\tilde{\mathbf{Y}}^2$

Equation (11) is rather simplistic in that it assumes a-priori that the error residuals follow a stationary AR(1) process. This assumption might not be particularly realistic for real-world studies. Various authors have therefore proposed alternative formulations of the likelihood function to extend applicability to situations where the error residuals are non-Gaussian with varying degrees of kurtosis and skewness (*Schoups and Vrugt*, 2010a; *Smith et al.*, 2010; *Evin et al.*, 2013; *Scharnagl et al.*, 2015). Latent variables can also be used to augment likelihood functions and take better consideration of forcing data and model structural error (*Kavetski et al.*, 2006a; *Vrugt et al.*, 2008a; *Renard et al.*, 2011). For systems with generative (negative) feedbacks, the error in the initial states poses no harm as its effect on system simulation rapidly diminishes when time advances. One can therefore take advantage of a spin-up period to remove sensitivity of the modeling results (and error residuals) to state value initialization.

The process of investigating phenomena, acquiring new information through experimentation and data collection, and refining existing theory and knowledge through Bayesian analysis has many elements in common with the scientific method. This framework, graphically illustrated in Figure 2 is adopted in many branches of the earth sciences, and seeks to elucidate the rules that govern the natural world.

---

<sup>2</sup>A nuisance variable is a random variable that is fundamental to the probabilistic model, but that is not of particular interest itself.

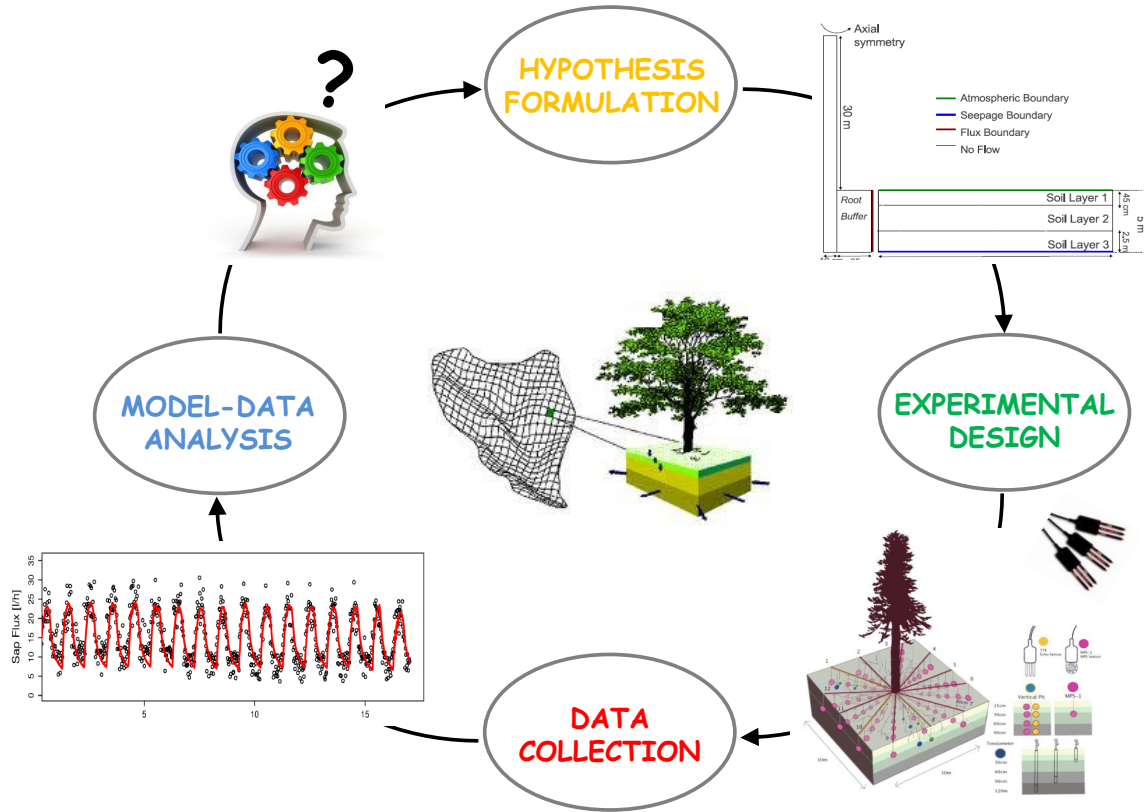


Figure 2 The iterative research cycle for a soil-tree-atmosphere-continuum (STAC). The initial hypothesis is that this system can be described accurately with a coupled soil-tree porous media model which simulates, using PDEs, processes such as infiltration, soil evaporation, variably saturated soil water flow and storage, root water uptake, xylem water storage and sapflux, and leaf transpiration. Measurements of spatially distributed soil moisture and matric head, sapflux, and tree trunk potential are used for model calibration and evaluation. The model-data comparison step reveals a systematic deviation in the early afternoon and night time hours between the observed (black circles) and simulated (solid red line) sapflux data. It has proven to be very difficult to pinpoint this epistemic error to a specific component of the model. Ad-hoc decisions on model improvement therefore usually prevail.

Once the prior distribution and likelihood function have been defined, what is left in Bayesian analysis is to summarize the posterior distribution, for example by the mean, the covariance or percentiles of individual parameters and/or nuisance variables. Unfortunately, most dynamic system models are highly nonlinear, and this task cannot be carried out by analytical means nor by analytical approximation. Confidence intervals construed from a classical first-order approximation can then only provide an approximate estimate of the posterior distribution. What is more, the target is assumed to be multivariate Gaussian ( $\ell^2$ -norm type likelihood function), a restrictive assumption. I therefore resort to Monte Carlo (MC) simulation methods to generate a sample of the posterior distribution.

In a previous paper, we have introduced the **D**iffe**R**ential **E**volution **A**ddaptive **M**etropolis (DREAM) algorithm (Vrugt *et al.*, 2008a, 2009a). This multi-chain Markov chain Monte Carlo (MCMC) simulation algorithm automatically tunes the scale and orientation of the



proposal distribution en route to the target distribution, and exhibits excellent sampling efficiencies on complex, high-dimensional, and multi-modal target distributions. DREAM is an adaptation of the Shuffled Complex Evolution Metropolis (*Vrugt et al.*, 2003) algorithm and has the advantage of maintaining detailed balance and ergodicity. Benchmark experiments [e.g. (*Vrugt et al.*, 2008a, 2009a; *Laloy and Vrugt*, 2012a; *Laloy et al.*, 2013; *Linde and Vrugt*, 2013; *Lochbühler et al.*, 2014; *Laloy et al.*, 2015)] have shown that DREAM is superior to other adaptive MCMC sampling approaches, and in high-dimensional search/variable spaces even provides better solutions than commonly used optimization algorithms.

In just a few years, the DREAM algorithm has found widespread application and use in numerous different fields, including (among others) atmospheric chemistry (*Partridge et al.*, 2011, 2012), biogeosciences (*Scharnagl et al.*, 2010; *Braakhekke et al.*, 2013; *Ahrens and Reichstein*, 2014; *Dumont et al.*, 2014; *Starrfelt and Kaste*, 2014), biology (*Coehlo et al.*, 2011; *Zaoli et al.*, 2014), chemistry (*Owejan et al.*, 2012; *Tarasevich et al.*, 2013; *DeCaluwe et al.*, 2014; *Gentsch et al.*, 2014), ecohydrology (*Dekker et al.*, 2011), ecology (*Barthel et al.*, 2011; *Gentsch et al.*, 2014; *Iizumi et al.*, 2014; *Zilliox and Goselin*, 2014), economics and quantitative finance (*Bauwens et al.*, 2011; *Lise et al.*, 2012; *Lise*, 2013), epidemiology (*Mari et al.*, 2011; *Rinaldo et al.*, 2012; *Leventhal et al.*, 2013), geophysics (*Bikowski et al.*, 2012; *Linde and Vrugt*, 2013; *Laloy et al.*, 2012b; *Carbajal et al.*, 2014; *Lochbühler et al.*, 2014, 2015), geostatistics (*Minasny et al.*, 2011; *Sun et al.*, 2013), hydrogeophysics (*Hinnell et al.*, 2014), hydrogeology (*Keating et al.*, 2010; *Laloy et al.*, 2013; *Malama et al.*, 2013), hydrology (*Vrugt et al.*, 2008a, 2009a; *Shafii et al.*, 2014), physics (*Dura et al.*, 2014; *Horowitz et al.*, 2012; *Toyli et al.*, 2012; *Kirby et al.*, 2013; *Yale et al.*, 2013; *Krayer et al.*, 2014), psychology (*Turner and Sederberg*, 2012), soil hydrology (*Wöhling and Vrugt*, 2011), and transportation engineering (*Kow et al.*, 2012).

In this paper, I review the basic theory of Markov chain Monte Carlo (MCMC) simulation, and summarize the different components of the DREAM algorithm as implemented in DREAM Suite. This Windows program provides scientists and engineers with a comprehensive set of capabilities for Bayesian inference and posterior exploration. DREAM Suite implements multi-core computing (if user desires) and includes tools for convergence analysis of the sampled chain trajectories and post-processing of the results. Recent extensions of the program are described as well, and include (among others) built-in functionalities that enable use of informal likelihood functions (*Beven and Binley*, 1992; *Beven and Freer*, 2001), summary statistics (*Gupta et al.*, 2008), approximate Bayesian computation (*Nott et al.*, 2012; *Sadegh and Vrugt*, 2013, 2014), diagnostic model evaluation (*Vrugt and Sadegh*, 2013a), and the limits of acceptability framework (*Beven*, 2006; *Beven and Binley*, 2014).

These developments are in part a response to the emerging paradigm of model diagnostics using summary statistics of system behavior. Recent work has shown that such approach provides better guidance on model malfunctioning and related issues than the conventional residual-based paradigm (*Sadegh et al.*, 2015b; *Vrugt*, 2015b). The main capabilities of the DREAM toolbox are demonstrated using seven different case studies involving (for instance) bimodality, high-dimensionality, summary statistics, bounded parameter spaces, dynamic simulation models, formal/informal likelihood functions, diagnostic model evaluation, data assimilation, Bayesian model averaging, distributed computation, informative/noninformative prior distributions, and limits of acceptability. These example studies are easy to run and adapt and serve as templates for other inference problems.

The remainder of this manual is organized as follows. Section 2 reviews the basic theory of Monte Carlo sampling and MCMC simulation. This is followed in section 3 with a brief discussion of adaptive single and multi-chain MCMC methods. Here, I discuss the DREAM algorithm. This section is especially concerned with the input and output arguments of DREAM Suite and the various functionalities, capabilities, and options available to the user. Section 5 of this paper illustrates the practical application of DREAM Suite to seven different case studies. These examples involve a wide variety of problem features, and illustrate some of the main capabilities of the program. Finally, section 6 concludes this manual with a summary of the work presented herein.

## 2. POSTERIOR EXPLORATION

A key task in Bayesian inference is to summarize the posterior distribution. When this task cannot be carried out by analytical means nor by analytical approximation, Monte Carlo simulation methods can be used to generate a sample from the posterior distribution. The desired summary of the posterior distribution is then obtained from the sample. The posterior distribution, also referred to as the target or limiting distribution, is often high dimensional. A large number of iterative methods have been developed to generate samples from the posterior distribution. All these methods rely in some way on Monte Carlo simulation. The next sections discuss several different posterior sampling methods.

### 2.1. Monte Carlo simulation

Monte Carlo methods are a broad class of computational algorithms that use repeated random sampling to approximate some multivariate probability distribution. The simplest Monte Carlo method involves random sampling of the prior distribution. This method is known to be rather inefficient, which I can illustrate with a simple example. Lets consider a circle with unit radius in a square of size  $\mathbf{x} \in [-2, 2]^2$ . The circle (posterior distribution)

has an area of  $\pi$  and makes up  $\pi/16 \approx 0.196$  of the prior distribution. I can now use Monte Carlo simulation to estimate the value of  $\pi$ . I do so by randomly sampling  $N = 10,000$  values of  $\mathbf{x}$  from the prior distribution. The  $M$  samples of  $\mathbf{x}$  that fall within the circle are posterior solutions and indicated with the plus symbol in Figure 3. Samples that fall outside the circle are rejected and printed with a dot. The value of  $\pi$  can now be estimated using  $\pi \approx 16M/N$  which in this numerical experiment with  $N = 10,000$  samples equates to 3.0912.

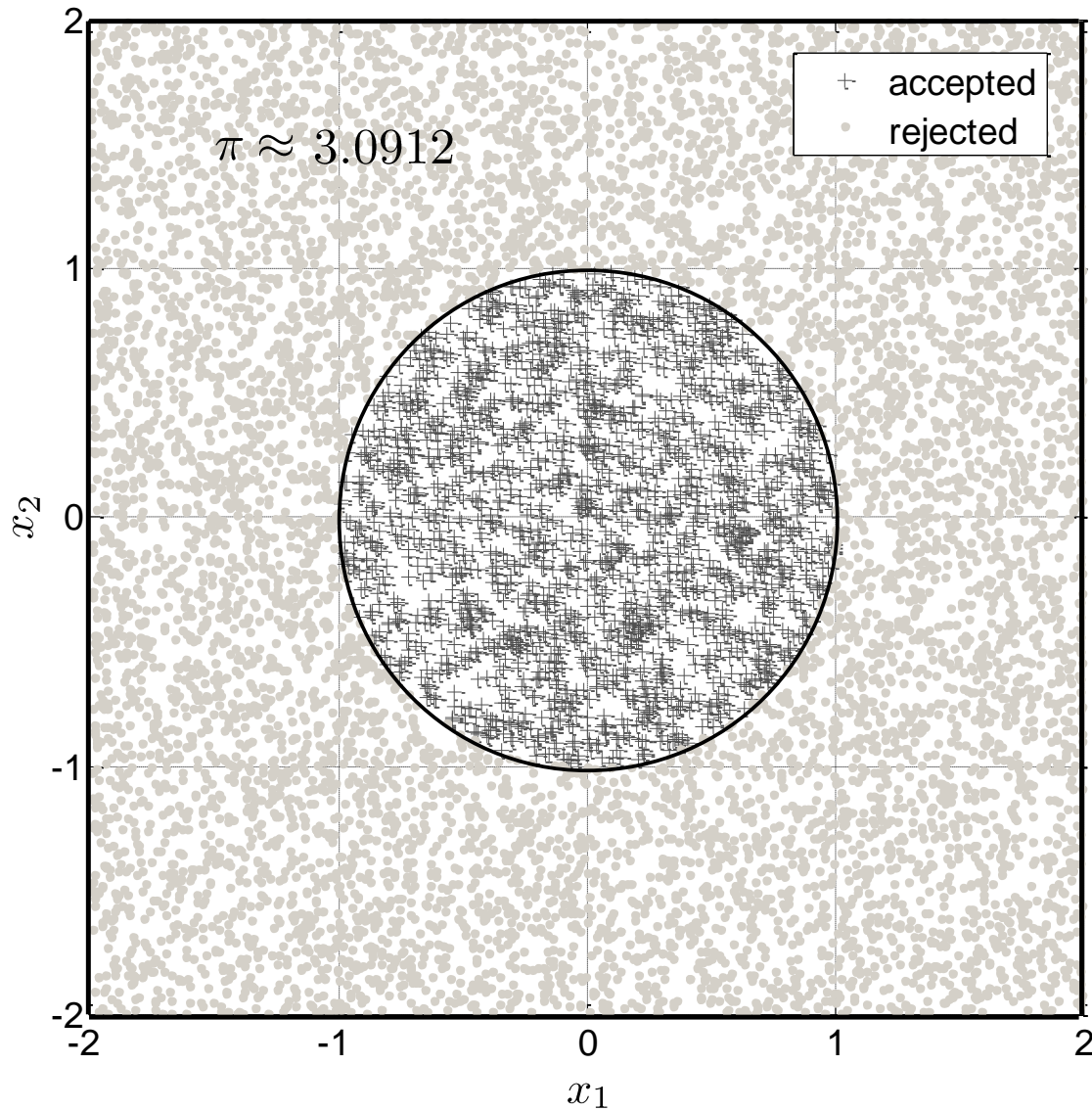


Figure 3 Example target distribution: A square with unit radius (in black) centered at the origin. The Monte Carlo samples are coded in dots (rejected) and plusses (accepted). The number of accepted samples can now be used to estimate the value of  $\pi \approx 3.0912$ .

The target distribution is relatively simple to sample in the present example. It should be evident however that uniform random sampling will not be particularly efficient if the hypercube of the prior distribution is much larger. Indeed, the chance that a random sample of  $\mathbf{x}$  falls within the unit circle decreases rapidly (quadratically) with increasing size of the prior distribution. If a much higher dimensional sample were considered then rejection sampling would quickly need many millions of Monte Carlo samples to delineate reasonably the posterior distribution and obtain an accurate value of  $\pi$ . What is more, in the present example all solutions within the circle have an equal density. If this were not the case then many accepted samples are required to approximate closely the distribution of the probability mass within the posterior distribution. Indeed, methods such as the generalized likelihood uncertainty estimation (GLUE) that rely on uniform sampling (such as rejection sampling) can produce questionable results if the target distribution is somewhat complex and/or comprises only a relatively small part of the prior distribution (Vrugt, 2015a). In summary, standard Monte Carlo simulation methods are computationally inefficient for anything but very low dimensional problems.

This example is rather simple but conveys what to expect when using simple Monte Carlo simulation methods to approximate complex and high-dimensional posterior distributions. I therefore resort to Markov chain Monte Carlo simulation to explore the posterior target distribution.

## 2.2. Markov Chain Monte Carlo simulation

The basis of MCMC simulation is a Markov chain that generates a random walk through the search space and successively visits solutions with stable frequencies stemming from a stationary distribution,  $\pi(\cdot)$ <sup>3</sup>. To explore the target distribution,  $\pi(\cdot)$ , a MCMC algorithm generates trial moves from the current state of the Markov chain  $\mathbf{x}_{t-1}$  to a new state  $\mathbf{x}_p$ . The earliest MCMC approach is the random walk Metropolis (RWM) algorithm introduced by Metropolis *et al.* (1953). This scheme is constructed to maintain detailed balance with respect to  $\pi(\cdot)$  at each step in the chain. If  $p(\mathbf{x}_{t-1})$  ( $p(\mathbf{x}_p)$ ) denotes the probability to find the system in state  $\mathbf{x}_{t-1}$  ( $\mathbf{x}_p$ ) and  $q(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p)$  ( $q(\mathbf{x}_p \rightarrow \mathbf{x}_{t-1})$ ) is the conditional probability to perform a trial move from  $\mathbf{x}_{t-1}$  to  $\mathbf{x}_p$  ( $\mathbf{x}_p$  to  $\mathbf{x}_{t-1}$ ), then the probability  $p_{\text{acc}}(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p)$  to accept the trial move from  $\mathbf{x}_{t-1}$  to  $\mathbf{x}_p$  is related to  $p_{\text{acc}}(\mathbf{x}_p \rightarrow \mathbf{x}_{t-1})$  according to

$$p(\mathbf{x}_{t-1})q(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p)p_{\text{acc}}(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p) = p(\mathbf{x}_p)q(\mathbf{x}_p \rightarrow \mathbf{x}_{t-1})p_{\text{acc}}(\mathbf{x}_p \rightarrow \mathbf{x}_{t-1}) \quad (12)$$

---

<sup>3</sup>This notation for the target distribution has nothing to do with the value of  $\pi = 3.1415\dots$  subject to inference in Figure 3

If a symmetric jumping distribution is used, that is  $q(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p) = q(\mathbf{x}_p \rightarrow \mathbf{x}_{t-1})$ , then it follows that

$$\frac{p_{\text{acc}}(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p)}{p_{\text{acc}}(\mathbf{x}_p \rightarrow \mathbf{x}_{t-1})} = \frac{p(\mathbf{x}_p)}{p(\mathbf{x}_{t-1})} \quad (13)$$

This Equation does not yet fix the acceptance probability. *Metropolis et al.* (1953) made the following choice

$$p_{\text{acc}}(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p) = \min \left[ 1, \frac{p(\mathbf{x}_p)}{p(\mathbf{x}_{t-1})} \right], \quad (14)$$

to determine whether to accept a trial move or not. This selection rule has become the basic building block of many existing MCMC algorithms. *Hastings* (1970) extended Equation (14) to the more general case of non-symmetrical jumping distributions

$$p_{\text{acc}}(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p) = \min \left[ 1, \frac{p(\mathbf{x}_p)q(\mathbf{x}_p \rightarrow \mathbf{x}_{t-1})}{p(\mathbf{x}_{t-1})q(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p)} \right], \quad (15)$$

in which the forward ( $\mathbf{x}_{t-1}$  to  $\mathbf{x}_p$ ) and backward ( $\mathbf{x}_p$  to  $\mathbf{x}_{t-1}$ ) jump do not have equal probability,  $q(\mathbf{x}_{t-1} \rightarrow \mathbf{x}_p) \neq q(\mathbf{x}_p \rightarrow \mathbf{x}_{t-1})$ . This generalization is known as the Metropolis-Hastings (MH) algorithm and broadens significantly the type of proposal distribution that can be used for posterior inference.

The core of the RWM algorithm can be coded in just a few lines and requires only a jumping distribution, a function to generate uniform random numbers, and a function to calculate the probability density of each proposal. Note, for the time being I conveniently assume the use of a noninformative prior distribution. This simplifies the Metropolis acceptance probability to the ratio of the densities of the proposal and the current state of the chain. The use of an informative prior distribution will be considered at a later stage.

In words, assume that the points  $\{\mathbf{x}_0, \dots, \mathbf{x}_{t-1}\}$  have already been sampled, then the RWM algorithm proceeds as follows. First, a candidate point  $\mathbf{x}_p$  is sampled from a proposal distribution  $q$  that depends on the present location,  $\mathbf{x}_{t-1}$  and is symmetric,  $q(\mathbf{x}_{t-1}, \mathbf{x}_p) = q(\mathbf{x}_p, \mathbf{x}_{t-1})$ . Next, the candidate point is either accepted or rejected using the Metropolis acceptance probability (Equation 14). Finally, if the proposal is accepted the chain moves to  $\mathbf{x}_p$ , otherwise the chain remains at its current location  $\mathbf{x}_{t-1}$ . Repeated application of these three steps results in a Markov chain which, under certain regularity conditions, has a unique stationary distribution with posterior probability density function,  $\pi(\cdot)$ . In practice, this means that if one looks at the values of  $\mathbf{x}$  sufficiently far from the arbitrary initial value, that is, after a burn-in period, the successively generated states of the chain will be distributed according to  $\pi(\cdot)$ , the posterior probability distribution of  $\mathbf{x}$ . Burn-in is required to allow the chain to explore the search space and reach its stationary regime.

Figure 4 illustrates the outcome of the RWM algorithm for a simple  $d = 2$ -dimensional multivariate normal target distribution with correlated dimensions. This target distribution is given by  $\psi_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , where  $\psi(\cdot)$  denotes the bivariate normal distribution with mean  $\boldsymbol{\mu} = \{\mu_1, \mu_2\}$  and covariance matrix  $\boldsymbol{\Sigma}$ . We assume a zero mean and  $d \times d$  covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & 0.8 \\ 0.9 & 1.0 \end{pmatrix}$$

The chain is initialized by sampling from  $\mathcal{U}_2[-10, 10]$ , where  $\mathcal{U}_d(a, b)$  denotes the  $d$ -variate uniform distribution with lower and upper bounds  $a$  and  $b$ , respectively. The left graph presents a scatter plot of the bivariate posterior samples using a total of  $T = 50,000$  function evaluations and burn-in of 50%. The contours depict the 68, 90, and 95% uncertainty intervals of the target distribution. The right graph displays a plot of the generation number against the value of parameter,  $x_1$  and  $x_2$  at each iteration. This is also called a traceplot.

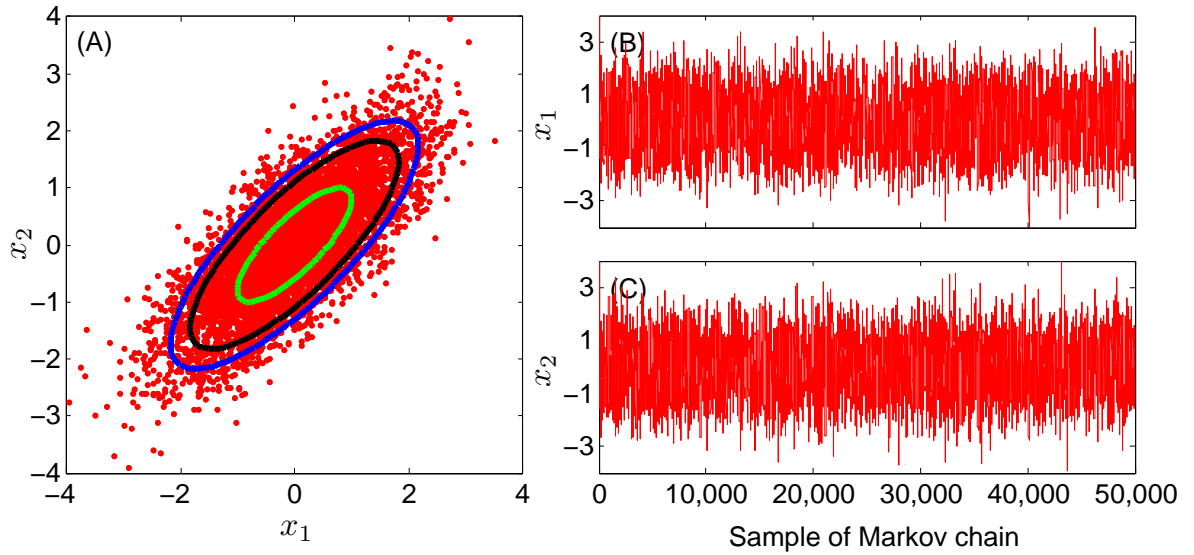


Figure 4 (A) bivariate scatter plots of the RWM derived posterior samples. The green, black and blue contour lines depict the true 68, 90 and 95% uncertainty intervals of the target distribution, respectively. (B,C) traceplot of the sampled values of  $x_1$  (top) and  $x_2$  (bottom).

Perhaps not surprisingly, the bivariate samples of the RWM algorithm nicely approximate the target distribution. The acceptance rate of 23% is somewhat lower than considered optimal in theory but certainly higher than derived from Monte Carlo simulation. The posterior mean and covariance are in excellent agreement with their values of the target distribution (not shown).

This simple example just serves to demonstrate the ability of RWM to approximate the posterior target distribution. The relative ease of implementation of RWM and its theoretical



underpinning have led to widespread application and use in Bayesian inference. However, the efficiency of the RWM algorithm is determined by the choice of the proposal distribution,  $q(\cdot)$  used to create trial moves (transitions) in the Markov chain. When the proposal distribution is too wide, too many candidate points are rejected, and therefore the chain will not mix efficiently and converge only slowly to the target distribution. On the other hand, when the proposal distribution is too narrow, nearly all candidate points are accepted, but the distance moved is so small that it will take a prohibitively large number of updates before the sampler has converged to the target distribution. The choice of the proposal distribution is therefore crucial and determines the practical applicability of MCMC simulation in many fields of study (*Owen and Tribble, 2005*).

### 3. AUTOMATIC TUNING OF PROPOSAL DISTRIBUTION

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.

#### 3.1. Single-chain methods

The most common adaptive single chain methods are the adaptive proposal (AP) (*Haario et al., 1999*), adaptive Metropolis (AM) (*Haario et al., 2001*) and delayed rejection adaptive metropolis (DRAM) algorithm (*Haario et al., 2006*), respectively. These methods work with a single trajectory, and continuously adapt the covariance,  $\Sigma$  of a Gaussian proposal distribution,  $q_t(\mathbf{x}_{t-1}, \cdot) = \mathcal{N}_d(\mathbf{x}_{t-1}, s_d \Sigma)$  using the accepted samples of the chain,  $\Sigma = \text{cov}(\mathbf{x}_0, \dots, \mathbf{x}_{t-1}) + \varphi \mathbf{I}_d$ . The variable  $s_d$  represents a scaling factor (scalar) that depends only on the dimensionality  $d$  of the problem,  $\mathbf{I}_d$  signifies the  $d$ -dimensional identity matrix, and  $\varphi = 10^{-6}$  is a small scalar that prevents the sample covariance matrix to become singular. As a basic choice, the scaling factor is chosen to be  $s_d = 2.38^2/d$  which is optimal for Gaussian target and proposal distributions (*Gelman et al., 1996; Roberts et al., 1997*) and should give an acceptance rate close to 0.44 for  $d = 1$ , 0.28 for  $d = 5$  and 0.23 for large  $d$ .

Single-site 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)* proposed to tune  $\Sigma$  using a limited-memory multi-block pre-sampling step, prior to a classical single block Metropolis run.

Another viable adaptation strategy is to keep the covariance matrix fixed (identity matrix) and to update during burn-in the scaling factor,  $s_d$  until a desired acceptance rate is achieved. This approach differs somewhat from the AM algorithm but is easy to implement.

Whether a specific adaptation scheme of the scaling factor (also called jump rate) works well in practice depends on the properties of the target distribution. Some tuning is hence required to achieve adequate results. Practical experience suggests that covariance matrix adaptation (AM) is preferred over scaling factor adaptation. The proposals created with the AM algorithm will more rapidly behave as the target distribution.

The use of a multivariate normal proposal distribution with adaptive covariance matrix or jump rate works well for Gaussian-shaped target distributions, but does not converge properly for multimodal distributions with long tails, possibly infinite first and second moments (as demonstrated in section . Experience further suggests that single chain methods are unable to traverse efficiently complex multi-dimensional parameter spaces with multiple different regions of attraction and numerous local optima. The use of an overly dispersed proposal distribution can help to avoid premature convergence, but with a very low acceptance rate in return. With a single chain it is also particularly difficult to judge when convergence has been achieved. Even the most powerful diagnostics that compare the sample moments of the first and second half of the chain cannot guarantee that the target distribution has been sampled. Indeed, the sample moments of both parts of the chain might be identical but the chain is stuck in a local optimum of the posterior surface or traverses consistently only a portion of the target distribution (*Gelman and Shirley, 2009*). In fact, single chain methods suffer many similar problems as local optimizers and cannot guarantee that the full parameter space has been explored adequately in pursuit of the target distribution.

### 3.2. Multi-chain methods: DE-MC

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, multi-modality, and numerous local optima (*Gilks et al., 1994; Liu et al., 2000; ter Braak, 2006; ter Braak and Vrugt, 2008; Vrugt et al., 2009a; Radu et al., 2009*). The use of multiple chains offers a robust protection against premature convergence, and opens up the use of a wide arsenal of statistical measures to test whether convergence to a limiting distribution has been achieved (*Gelman and Rubin, 1992*). One popular multi-chain method that has found widespread application and use in hydrology is the Shuffled Complex Evolution Metropolis algorithm (SCEM-UA, *Vrugt et al., 2003*). Although the proposal adaptation of SCEM-UA violates Markovian properties, numerical benchmark experiments on a diverse set of multi-variate target distributions have shown that the method is efficient and close to exact. The difference between the limiting distribution of SCEM-UA and the true target distribution is negligible in most reasonable cases and applications. The SCEM-UA method



can be made an exact sampler if the multi-chain adaptation of the covariance matrix is restricted to the burn-in period only. In a fashion similar to the AP (*Haario et al.*, 1999) and AM algorithm, the method then derives an efficient Gaussian proposal distribution for the standard Metropolis algorithm. Nevertheless, I do not consider the SCEM-UA algorithm herein.

*ter Braak* (2006) proposed a simple adaptive RWM algorithm called Differential Evolution Markov chain (DE-MC). 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. In DE-MC,  $N$  different Markov chains are run simultaneously in parallel. If the state of a single chain is given by the  $d$ -vector  $\mathbf{x}$ , then at each generation  $t - 1$  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. Then multivariate proposals,  $\mathbf{X}_p$  are generated on the fly from the collection of chains,  $\mathbf{X} = \{\mathbf{x}_{t-1}^1, \dots, \mathbf{x}_{t-1}^N\}$  using differential evolution (*Storn and Price*, 1997; *Price et al.*, 2005)

$$\mathbf{X}_p^i = \gamma_d(\mathbf{X}^a - \mathbf{X}^b) + \boldsymbol{\zeta}_d, \quad a \neq b \neq i, \quad (16)$$

where  $\gamma$  denotes the jump rate,  $a$  and  $b$  are integer values drawn without replacement from  $\{1, \dots, i-1, i+1, \dots, N\}$ , and  $\boldsymbol{\zeta} \stackrel{\mathcal{D}}{\sim} \mathcal{N}_d(0, c_*)$  is drawn from a normal distribution with small standard deviation, say  $c_* = 10^{-6}$ . By accepting each proposal with Metropolis probability

$$p_{\text{acc}}(\mathbf{X}^i \rightarrow \mathbf{X}_p^i) = \min[1, p(\mathbf{X}_p^i)/p(\mathbf{X}^i)], \quad (17)$$

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). Thus, if  $p_{\text{acc}}(\mathbf{X}^i \rightarrow \mathbf{X}_p^i)$  is larger than some uniform label drawn from  $\mathcal{U}(0, 1)$  then the candidate point is accepted and the  $i$ th chain moves to the new position, that is  $\mathbf{x}_t^i = \mathbf{X}_p^i$ , otherwise  $\mathbf{x}_t^i = \mathbf{x}_{t-1}^i$ .

Because the joint pdf of the  $N$  chains factorizes to  $\bar{\pi}(\mathbf{x}^1|\cdot) \times \dots \times \bar{\pi}(\mathbf{x}^N|\cdot)$ , the states  $\mathbf{x}^1 \dots \mathbf{x}^N$  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). If the initial population is drawn from the prior distribution, then DE-MC translates this sample into a posterior population. From the guidelines of  $s_d$  in RWM the optimal choice of  $\gamma = 2.38/\sqrt{2d}$ . With a 10% probability the value of  $\gamma = 1$ , or  $p_{(\gamma=1)} = 0.1$  to allow for mode-jumping (*ter Braak*, 2006; *ter Braak and Vrugt*, 2008; *Vrugt et al.*, 2008a, 2009a) which is a significant strength of DE-MC as will be shown later. If the posterior distribution consists of disconnected modes with in-between regions of low probability, covariance based

MCMC methods will be slow to converge as transitions between probability regions will be infrequent.

The DE-MC method solves an important practical problem in RWM, 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.

To demonstrate the advantages of DE-MC over single chain methods please consider Figure 5 that presents histograms of the posterior samples derived from AM (left plot) and DE-MC (right plot) for a simple univariate target distribution consisting of a mixture of two normal distributions,

$$p(x) = \frac{1}{6}\psi(-8, 1) + \frac{5}{6}\psi(10, 1), \quad (18)$$

where  $\psi(a, b)$  denotes the probability density function (pdf) of a normal distribution with mean  $a$  and standard deviation  $b$ . The initial state of the Markov chain(s) is sampled from  $\mathcal{U}[-20, 20]$  (see Figure 5).

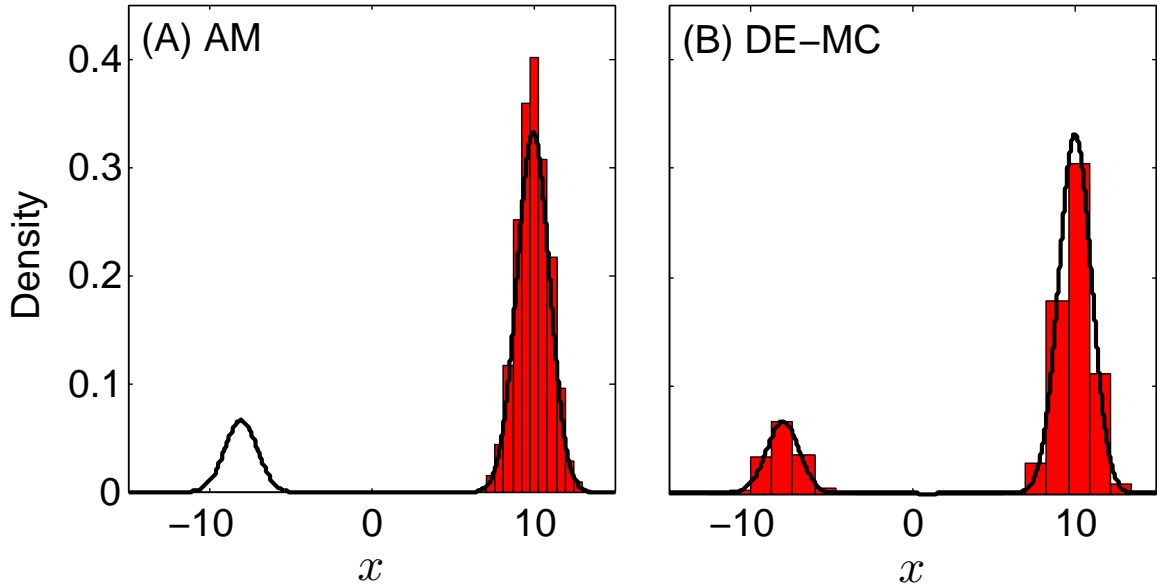


Figure 5 Histogram of the posterior distribution derived from the (A) AM (single chain), and (B) DE-MC (multi-chain) samplers. The solid black line displays the pdf of the true mixture target distribution.

The AM algorithm produces a spurious approximation of the bimodal target distribution. The variance (width) of the proposal distribution is insufficient to enable the chain to adequately explore both modes of the target distribution. A simple remedy to this problem is to increase the (default) initial variance of the univariate normal proposal distribution. This would allow the AM sampler to take much larger steps and jump directly between

both modes, but at the expense of a drastic reduction in the acceptance rate and search efficiency. Indeed, besides the occasional successful jumps many other proposals will overshoot the target distribution, receive a nearly zero density, and consequently be rejected.

This rather simple univariate example illustrates the dilemma of RWM how to determine an appropriate scale and orientation of the proposal distribution. Fortunately, the histogram of the posterior samples derived with the DE-MC algorithm matches perfectly the mixture distribution. Periodic use of  $\gamma = 1$  enables the  $N = 10$  different Markov chains of DE-MC to transition directly between the two disconnected posterior modes (e.g. *ter Braak and Vrugt* (2008); *Vrugt et al.* (2008a); *Laloy and Vrugt* (2012a)) and rapidly converge to the exact target distribution. The initial states of the DE-MC chains should be distributed over the parameter space so that both modes can be found. What is more the use of  $N$  trajectories allows for a much more robust assessment of convergence.

In previous work (*Vrugt et al.*, 2008a, 2009a) we have shown that the efficiency of DE-MC can be enhanced, sometimes dramatically, by using adaptive randomized subspace sampling, multiple chain pairs for proposal creation, and explicit consideration of aberrant trajectories. This method, entitled **D**iffe**R**ential **E**volution **A**ddaptive **M**etropolis (DREAM) maintains detailed balance and ergodicity and has shown to exhibit excellent performance on a wide range of problems involving nonlinearity, high-dimensionality, and multimodality. In these and other papers [e.g (*Laloy and Vrugt*, 2012a)] benchmark experiments have shown that DREAM outperforms other adaptive MCMC sampling approaches, and, in high-dimensional search/variable spaces, can even provide better solutions than commonly used optimization algorithms.

### 3.3. Multi-chain methods: DREAM

The DREAM algorithm has its roots within DE-MC but uses subspace sampling and outlier chain correction to speed up convergence to the target distribution. Subspace sampling is implemented in DREAM by only updating randomly selected dimensions of  $\mathbf{x}$  each time a proposal is generated. If  $A$  is a subset of  $d^*$ -dimensions of the original parameter space,  $\mathbb{R}^{d^*} \subseteq \mathbb{R}^d$ , then a jump,  $d\mathbf{X}^i$  in the  $i$ th chain,  $i = \{1, \dots, N\}$  at iteration  $t = \{2, \dots, T\}$  is calculated from the collection of chains,  $\mathbf{X} = \{\mathbf{x}_{t-1}^1, \dots, \mathbf{x}_{t-1}^N\}$  using differential evolution (*Storn and Price*, 1997; *Price et al.*, 2005)

$$\begin{aligned} d\mathbf{X}_A^i &= \zeta_{d^*} + (\mathbf{1}_{d^*} + \lambda_{d^*}) \gamma_{(\delta, d^*)} \sum_{j=1}^{\delta} (\mathbf{X}_A^{\mathbf{a}_j} - \mathbf{X}_A^{\mathbf{b}_j}) \\ d\mathbf{X}_{\neq A}^i &= 0, \end{aligned} \tag{19}$$

where  $\gamma = 2.38 / \sqrt{2\delta d^*}$  is the jump rate,  $\delta$  denotes the number of chain pairs used to generate the jump, and  $\mathbf{a}$  and  $\mathbf{b}$  are vectors consisting of  $\delta$  integers drawn without replacement from  $\{1, \dots, i-1, i+1, \dots, N\}$ . The default value of  $\delta = 3$ , and results, in practice, in one-third of the proposals being created with  $\delta = 1$ , another third with  $\delta = 2$ , and the remaining third using  $\delta = 3$ . The values of  $\lambda$  and  $\zeta$  are sampled independently from  $\mathcal{U}_{d^*}(-c, c)$  and  $\mathcal{N}_{d^*}(0, c_*)$ , respectively, the multivariate uniform and normal distribution with, typically,  $c = 0.1$  and  $c_*$  small compared to the width of the target distribution,  $c_* = 10^{-6}$  say. Compared to DE-MC,  $p_{(\gamma=1)} = 0.2$  to enhance the probability of jumps between disconnected modes of the target distribution. The candidate point of chain  $i$  at iteration  $t$  then becomes

$$\mathbf{X}_p^i = \mathbf{X}^i + \mathbf{dX}^i, \quad (20)$$

and the Metropolis ratio of Equation (17) is used to determine whether to accept this proposal or not. If  $p_{\text{acc}}(\mathbf{X}^i \rightarrow \mathbf{X}_p^i) \geq \mathcal{U}(0, 1)$  the candidate point is accepted and the  $i$ th chain moves to the new position, that is  $\mathbf{x}_t^i = \mathbf{X}_p^i$ , otherwise  $\mathbf{x}_t^i = \mathbf{x}_{t-1}^i$ . The default equation for  $\gamma$  should, for Gaussian and Student target distribution, result in optimal acceptance rates close to 0.44 for  $d = 1$ , 0.28 for  $d = 5$ , and 0.23 for large  $d$  (please refer to section 7.84 of *Roberts and Casella* (2004) for a cautionary note on these references acceptance rates).

The  $d^*$ -members of the subset A are sampled from the entries  $\{1, \dots, d\}$  (without replacement) and define the dimensions of the parameter space to be sampled by the proposal. This subspace spanned by A is construed in DREAM with the help of a crossover operator. This genetic operator is applied before each proposal is created and works as follows. First, a crossover value, cr is sampled from a geometric sequence of  $n_{\text{CR}}$  different crossover probabilities,  $\text{CR} = \{\frac{1}{n_{\text{CR}}}, \frac{2}{n_{\text{CR}}}, \dots, 1\}$  using the discrete multinomial distribution,  $\mathcal{M}(\text{CR}, \mathbf{p}_{\text{CR}})$  on CR with selection probabilities  $\mathbf{p}_{\text{CR}}$ . Then, a  $d$ -vector  $\mathbf{z} = \{z_1, \dots, z_d\}$  is drawn from a standard multivariate normal distribution,  $\mathbf{z} \stackrel{\mathcal{D}}{\sim} \mathcal{U}_d(0, 1)$ . All those values  $j$  which satisfy  $z_j \leq \text{cr}$  are stored in the subset A and span the subspace of the proposal that will be sampled using Equation (19). If A is empty, one dimension of  $\{1, \dots, d\}$  will be sampled at random to avoid the jump vector to have zero length.

The use of a vector of crossover probabilities enables single-site Metropolis (A has one element), Metropolis-within-Gibbs (A has one or more elements) and regular Metropolis sampling (A has  $d$  elements), and constantly introduces new directions in the parameter space that chains can take outside the subspace spanned by their current positions. What is more, the use of subspace sampling allows using  $N < d$  in DREAM, an important advantage over DE-MC that requires  $N = 2d$  chains to be run in parallel (*ter Braak*, 2006). Subspace sampling as implemented in DREAM adds one extra algorithmic variable,  $n_{\text{CR}}$

to the algorithm. The default setting of  $n_{\text{CR}} = 3$  has shown to work well in practice, but larger values of this algorithmic variable might seem appropriate for high-dimensional target distributions, say  $d > 50$ , to preserve the frequency of low-dimensional jumps. Note, more intelligent subspace selection methods can be devised for target distributions involving many highly correlated parameters. These parameters should be sampled jointly in a group, otherwise too many of the (subspace) proposals will be rejected and the search can stagnate. This topic will be explored in future work.

To enhance search efficiency the selection probability of each crossover value, stored in the  $n_{\text{CR}}$ -vector  $\mathbf{p}_{\text{CR}}$ , is tuned adaptively during burn-in by maximizing the distance traveled by each of the  $N$  chains. This adaptation is described in detail in *Vrugt et al.* (2008a, 2009a).

The DREAM algorithm described above differs in several important ways from the DE-MC algorithm presented in section 3.2. These added features enhance significantly the convergence speed of the sampled chains to a limiting distribution. The DREAM algorithm does implement outlier handling to patch a critical vulnerability of multi-chain MCMC methods such as SCEM-UA, DE-MC, and DREAM (*Vrugt et al.*, 2003; *ter Braak and Vrugt*, 2008; *Vrugt et al.*, 2008a, 2009a). The performance of these methods is impaired if one or more of their sampled chains have become trapped in an unproductive area of the parameter space while in pursuit of the target distribution. The state of these outlier chains will not only contaminate the jumping distribution of Equation (19) and thus slow down the evolution and mixing of the other "good" chains, what is much worse, dissident chains make it impossible to reach convergence to a limiting distribution. For as long as one of the chains samples a disjoint part of the parameter space, the  $\hat{R}$ -diagnostic of *Gelman and Rubin* (1992) cannot reach its stipulated threshold of 1.2 required to officially declare convergence.

The problem of outlier chains is well understood and easily demonstrated with an example involving a posterior response surface with one or more local area of attractions far removed from the target distribution. Chains that populate such local optima can continue to persist indefinitely if the size of their jumps is insufficient to move the chain outside the space spanned by this optima (see Figure 2 of *ter Braak and Vrugt* (2008)). Dissident chains will occur most frequent in high-dimensional target distributions, as they require the use of a large  $N$ , and complex posterior response surfaces with many areas of attraction.

DREAM suite implements a function called **check** to remedy dissident chains. The mean log density of the samples stored in the second half of each chain is used as proxy for the "fitness" of each trajectory, and these  $N$  data points are examined for anomalies using an outlier detection test. Those chains (data points) that have been labeled as outlier will relinquish their dissident state and move to the position of one of the other chains (chosen

at random). Details of this procedure can be found in *Vrugt et al. (2009a)*. DREAM Suite implements four different outlier detection methods the user can choose from. Details will be presented in the next section.

Those proficient in statistics, computer coding and numerical computation, will be able to code the DREAM algorithm and personalize this method for their own applications. Yet, for others we have developed a C++ program of the algorithm. This program is called DREAM Suite and implements many built-in functionalities that are easy to use in practice. The next sections will introduce the various elements of DREAM Suite, and use several examples to illustrate how the package can be used to solve a wide variety of Bayesian inference problems involving (among others) simple functions, dynamic simulation models, formal and informal likelihood functions, informative and noninformative prior distributions, limits of acceptability, summary statistics, diagnostic model evaluation, low and high-dimensional parameter spaces, and distributed computing.

Numerical experiments with a large and diverse set of test functions have shown that the parallel implementation of DREAM Suite enhances convergence to the posterior distribution for numerical models that require at least 2 seconds to run.

#### 4. IMPLEMENTATION OF DREAM

The source code of DREAM Suite was originally written in MATLAB in 2006 and many new functionalities and options have been added to the source code in recent years due to continued research developments and to support the needs of a growing group of users. In 2015 we have developed a C++ code of the MATLAB toolbox, which is called DREAM Suite. This version is compiled for Windows and has a Graphical User Interface (GUI) to simplify implementation and use of the DREAM algorithm.

The DREAM code can be executed by the user from the GUI. This requires the user to specify the input arguments of DREAM. I now discuss all these input variables - that can be defined in the different tabs of the GUI. If the main function call to DREAM is given by

```
[chain,output,fx] = DREAM(model,DREAMPar,Par_info,Meas_info),options
```

then the input arguments are `model` (plugin written by user), `DREAMPar`, `Par_info`, `Meas_info` and `options` are input arguments defined by the user in the GUI, and `chain`, `output` and `fx` are output variables computed by the C++ function of DREAM and returned to the user in binary or text files and their content is visualized in tables and figures by the postprocessor of DREAM Suite. I will now discuss each of the five input variables and their content and usage.

#### 4.1. Input argument 1: *model*

The variable `model` is perhaps the most important input argument of the C++ code as this constitutes the model or function plugin written by the user. `model` could be as simple as a simple statistical distribution that is used to test DREAM, or as complex as a CPU-intensive numerical model that solves differential equations in time and space. I conveniently write the `model` operator as follows

$$\mathbf{Y} = \text{model}(\mathbf{x}) \quad (21)$$

where  $\mathbf{x}$  (input argument) is a  $1 \times d$  vector of parameter values sampled by DREAM, and  $\mathbf{Y}$  is a return argument whose content is either a likelihood, log-likelihood, or vector of simulated values or summary statistics, respectively. The content of the function `model` needs to be written by the user - but the syntax and function call is universal. The user is responsible for their own plugin. The DREAM suite manual clarifies how to setup the `model` program with DREAM Suite.

#### 4.2. Input argument 1: *DREAMPar*

The DREAM algorithm has a number of algorithmic variables that determine the performance and efficiency of the algorithm. Some of these variables are case study dependent, whereas others should be kept at their default values. Table 1 summarizes the algorithmic variables of DREAM.

Table 1 Main algorithmic variables of DREAM: Mathematical symbols, corresponding fields in DREAM Suite, and default settings. Note, the default settings have been determined in previous work and shown to work well for a range of target distributions.

| Symbol                                       | Description                    | Field DREAMPar | Default                                 |
|--|--------------------------------|----------------|---|
| Problem dependent                            |                                |                |   |
| $d$  | number of parameters           | d              | $\geq 1$                                |
| $N$  | number of Markov chains        | N              | $\leq 2\delta + 1$                      |
| $T$  | number of generations          | T              | $\geq 1$                                |
| $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}})$ | likelihood function            | lik            | [1, 2], [11 – 17], [21 – 23], [31 – 34] |
| Default variables <sup>†</sup>               |                                |                |   |
| $n_{cr}$                                     | number of crossover values     | nCR            | 3                                       |
| $\delta$                                     | number chain pairs proposal    | delta          | 3                                       |
| $\lambda$ <sup>‡</sup>                       | randomization                  | lambda         | 0.1                                     |
| $\zeta$ <sup>§</sup>                         | ergodicity                     | zeta           | $10^{-12}$                              |
| $p(\gamma=1)$                                | probability unit jump rate     | p_unit_gamma   | 0.2                                     |
|  | outlier detection test         | outlier        | 'iqr'                                   |
| $K$  | thinning rate                  | thinning       | 1                                       |
|  | adapt crossover probabilities? | adapt_pCR      | 'yes'                                   |
| $G$ <sup>ℒ</sup>                             | shaping factor                 | GLUE           | $> 0$                                   |
| $\beta_0$ <sup>¶</sup>                       | scaling factor jump rate       | beta0          | 1                                       |

<sup>†</sup> A change to the default values of DREAM will affect the convergence (acceptance) rate

<sup>‡</sup>  $\lambda \stackrel{\mathcal{D}}{\sim} \mathcal{U}_{d*}(-\text{lambda}, \text{lambda})$

<sup>§</sup>  $\zeta \stackrel{\mathcal{D}}{\sim} \mathcal{N}_{d*}(0, \text{zeta})$

<sup>ℒ</sup> For pseudo-likelihood functions of GLUE (*Beven and Binley, 1992*)

<sup>¶</sup> Multiplier of the jump rate,  $\gamma = \beta_0\gamma$ , default  $\beta_0 = 1$

The values of the algorithmic parameters stored in DREAMPar can be defined by the user in the GUI of DREAM Suite. Most fields of DREAMPar appear in the "DREAM algorithm" tab.



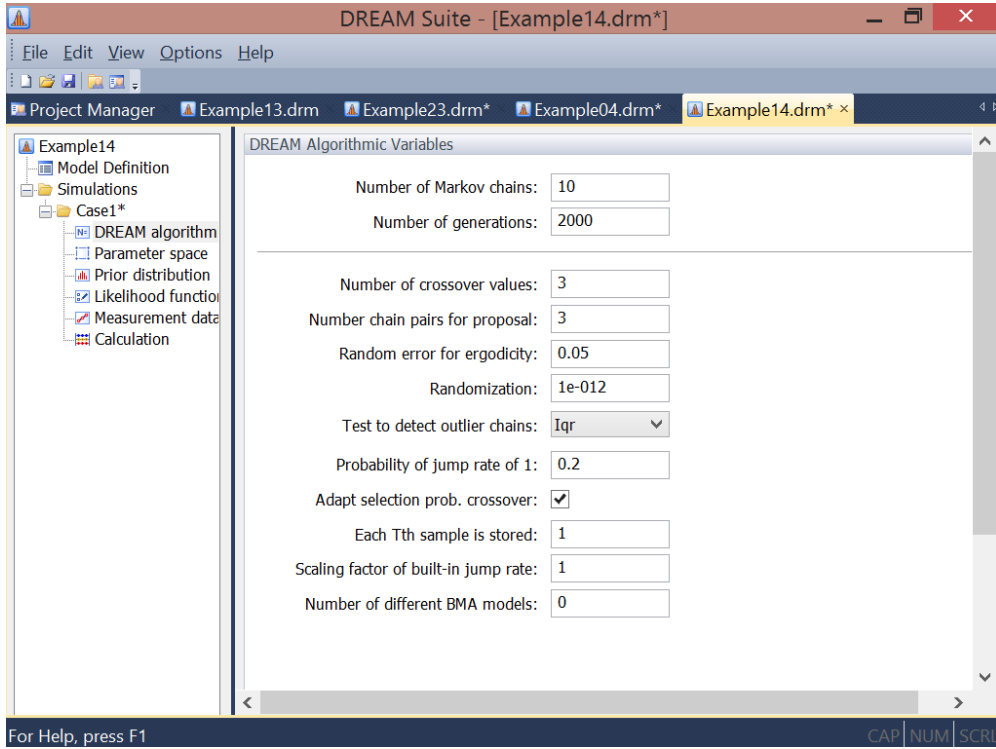


Figure 6 Screenshot of "DREAM algorithm" tab of the GUI of DREAM Suite. Note that the field "Number of different BMA models" applies only to example 19 of the program manager

The field `d` of `DREAMPar` appears on the "Parameter space" tab of the GUI, and the field `lik` and `G` on the "Likelihood function" tab, of which more later.

The field names of `DREAMPar` match exactly the symbols (letters) used in the (mathematical) description of DREAM in Equations (19) and (20). The values of the fields `d`, `N`, `T` depend on the dimensionality of the target distribution. These variables are problem dependent and should hence be specified by the user in the GUI. Default settings are assumed in Table 1 for the remaining fields of `DREAMPar` with the exception of `GLUE` and `lik` whose values will be discussed in the next two paragraphs. To create proposals with Equation (19), the value of `N` should at least be equivalent to  $2\delta + 1$  or  $N = 7$  for the default of  $\delta = 3$ . This number of chains is somewhat excessive for low dimensional problems involving just a few parameters. One could therefore conveniently set  $\delta = 1$  for small  $d$ . The default settings of `DREAMPar` are easy to modify by the user in the GUI, yet this is generally not recommended.

The DREAM algorithm can be used to sample efficiently the behavioral solution space of informal and likelihood functions used within GLUE (Beven and Binley, 1992; Beven and Freer, 2001). In fact, as will be shown later, DREAM can also solve efficiently the limits of acceptability framework of Beven (2006). For now it suffices to say that the field `GLUE` of `DREAMPar` stores the value of the shaping factor used within the (pseudo)likelihood

functions of GLUE. I will revisit GLUE and informal Bayesian inference at various places in the remainder of this manual. The content of the field `lik` of `DREAMPar` defines the choice of likelihood function used to compare the output of the model plugin with the available calibration data. Table 2 lists the different options for `lik` the user can select from. These likelihood functions appear under the "Likelihood function" tab of the GUI. The choice of likelihood function depends in large part on the content of the return argument `Y` of the users model plugin which is either a (log)-likelihood, a vector with simulated values, or a vector with summary statistics, respectively.

Table 2 Built-in likelihood functions of DREAM Suite. The value of field `lik` of `DREAMPar` depends on the content of the return argument `Y` from the model plugin of the user; [1] likelihood, [2] log-likelihood, [11-17] vector of simulated values, [21-23] vector of summary statistics, and [31-34] vector of simulated values. The mathematical formulation of each likelihood function is given in Appendix A. The likelihood functions appear under the "Likelihood function" tab of the GUI.

| lik                                  | Description  | References                                     |
|--------------------------------------|--|--|
| User-free likelihood functions       |  |  |
| 1                                    | Likelihood, $L(\mathbf{x} \tilde{\mathbf{Y}})$               | e.g. Equation (7)                              |
| 2                                    | Log-likelihood, $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}})$ | e.g. Equations (8), (10) and (11)              |
| Formal likelihood functions          |  |  |
| 11                                   | Gaussian likelihood: measurement error integrated out        | <i>Thiemann et al. (2001)</i> ; see footnote 1 |
| 12 †                                 | Gaussian likelihood: homos/heteroscedastic data error        | Equation (7)                                   |
| 13 †‡                                | Gaussian likelihood: with AR-1 model of error residuals      | Equations (10) and (11)                        |
| 14 §                                 | Generalized likelihood function                              | <i>Schoups and Vrugt (2010a)</i>               |
| 15                                   | Whittle's likelihood (spectral analysis)                     | <i>Whittle (1953)</i>                          |
| 16 †                                 | Laplacian likelihood: homos/heteroscedastic data error       | <i>Laplace (1774)</i>                          |
| 17 §                                 | Skewed Student likelihood function                           | <i>Scharnagl et al. (2015)</i>                 |
| ABC - diagnostic model evaluation    |  |  |
| 21 ¶                                 | Noisy ABC: Gaussian likelihood                               | <i>Turner and Sederberg (2012)</i>             |
| 22 ¶¥                                | ABC: Boxcar likelihood                                       | <i>Sadegh and Vrugt (2014)</i>                 |
| GLUE - limits of acceptability       |  |  |
| 23 ¥                                 | Limits of acceptability                                      | <i>Vrugt (2015a)</i>                           |
| GLUE - informal likelihood functions |  |  |
| 31 £                                 | Inverse error variance with shaping factor                   | <i>Beven and Binley (1992)</i>                 |
| 32 £                                 | Nash and Sutcliffe efficiency with shaping factor            | <i>Freer et al. (1996)</i>                     |
| 33 £                                 | Exponential transform error variance with shaping factor     | <i>Freer et al. (1996)</i>                     |
| 34 £                                 | Sum of absolute error residuals                              | <i>Beven and Binley (1992)</i>                 |

† Measurement data error in field `Sigma` of `Par_info` or inferred jointly with parameters

‡ First-order autoregressive coefficient is nuisance variable

§ Nuisance variables for model bias, correlation, non-stationarity and nonnormality residuals

¶ € delineates behavioral space: see "Summary Statistics" tab in GUI

¥ Uses a modified Metropolis selection rule to accept proposals or not (see Page 41)

£ Shaping factor,  $G$  defined in field `GLUE` of `DREAMPar` (default  $G = 10$ )

The likelihood function can be chosen by the user in the GUI of DREAM Suite under the "Likelihood function" tab.

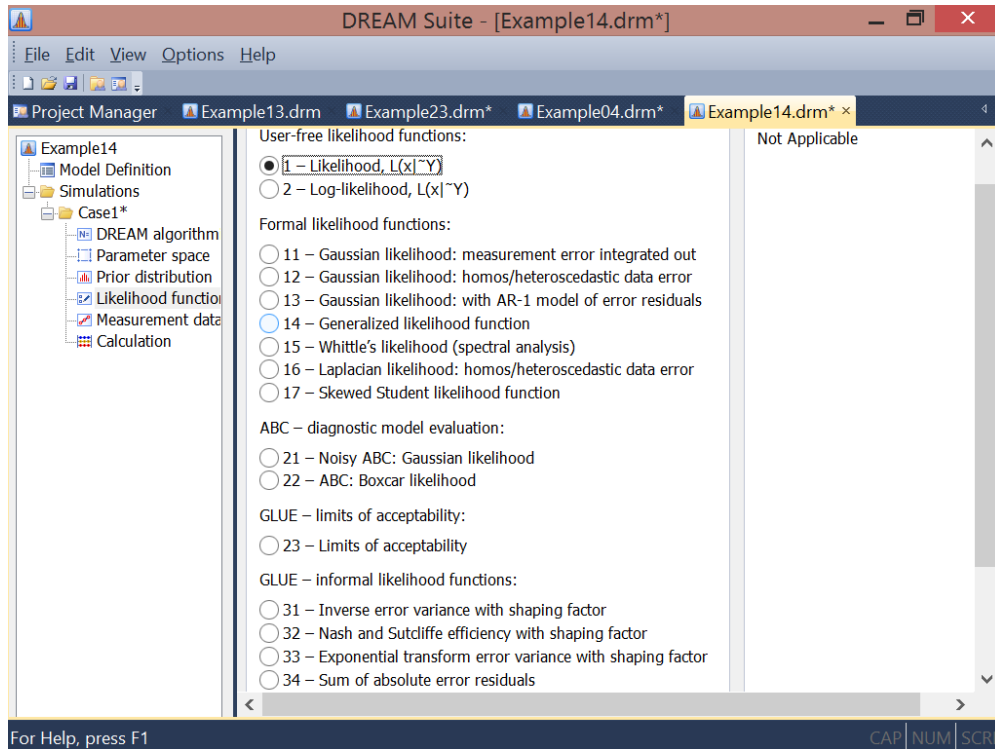


Figure 7 Screenshot of "Likelihood function" tab of the GUI of DREAM Suite.

If the return argument,  $Y$  of the model plugin of the user, called `model` is equivalent to a likelihood or log-likelihood then field `lik` of `DREAMPar` should be set equivalent to 1 or 2, respectively. This choice is appropriate for problems involving some prescribed multivariate probability distribution whose density can be evaluated directly. Examples of such functions are presented in project manager of the GUI of DREAM suite (example 1, 2, 3, 8, 10, 11). Option 1 and 2 also enable users to evaluate their own preferred likelihood function directly in the `model` script. In principle, these two options are therefore sufficient to apply the DREAM code to a large suite of problems. Nevertheless, to simplify implementation and use, the DREAM package contains 14 different built-in likelihood functions.

Likelihood functions 11-17 and 31-34 are appropriate if the output of `model` plugin consists of a vector of simulated values of some variable(s) of interest (see examples 4-7, 9, 12-14 and 16-19 of the project manager of the GUI of DREAM suite). Some of these likelihood functions (e.g., 12-14, 16, 17) contain extraneous variables (nuisance coefficients) whose values need to be inferred jointly with the model parameters,  $\mathbf{x}$  (see examples 14 and 17 of the project manager of the GUI of DREAM suite). Likelihood functions 21 and 22 are appropriate if the return argument  $Y$  of `model` plugin consists of one or more summary statistics of the simulated data (see examples 8 and 14 of the project manager of the

GUI of DREAM suite). These two likelihood functions allow use of approximate Bayesian computation and diagnostic model evaluation (Vrugt and Sadegh, 2013a; Sadegh and Vrugt, 2014; Vrugt, 2015b). Finally, likelihood function 23 enables use of the limits of acceptability framework (Beven, 2006; Beven and Binley, 2014; Vrugt, 2015a) (see examples 20, 21, 22, 23 of the project manager of the GUI of DREAM suite). Appendix A provides the mathematical formulation of each of the likelihood functions listed in Table 2. Note, likelihood 22 and 23 use a modified Metropolis selection rule to accept proposals or not (see Page 41).

The generalized likelihood (GL) function of Schoups and Vrugt (2010a) (14) is most advanced in that it can account explicitly for bias, correlation, non-stationarity, and non-normality of the error residuals through the use of nuisance coefficients. In a recent paper, Scharnagl et al. (2015) has introduced a skewed student likelihood function (17) as modification to the GL formulation (14) to describe adequately heavy-tailed error residual distributions. Whittle's likelihood (Whittle, 1953) (15) is a frequency-based approximation of the Gaussian likelihood and can be interpreted as minimum distance estimate of the distance between the parametric spectral density and the (nonparametric) periodogram (see examples 9 of the project manager of the GUI of DREAM suite). It also minimizes the asymptotic Kullback-Leibler divergence and, for autoregressive processes, provides asymptotically consistent estimates for Gaussian and non-Gaussian data, even in the presence of long-range dependence (Montanari and Toth, 2007). Likelihood function 16, also referred to as Laplace or double exponential distribution, differs from all other likelihood functions in that it assumes a  $\ell^1$ -norm of the error residuals. This approach weights all error residuals equally and the posterior inference should therefore not be as sensitive to outliers.

Likelihood functions 11-17 and 31-34 represent a different school of thought. Formulations 11-17 are derived from first-order statistical principles about the expected probabilistic properties of the error residuals,  $\mathbf{E}(\mathbf{x}) = \tilde{\mathbf{Y}} - \mathbf{Y}(\mathbf{x})$ . These functions are also referred to as formal likelihood functions. For example if the error residuals are assumed to be independent (uncorrelated) and normally distributed then the likelihood function is simply equivalent to formulation 11 or 12, depending on whether the measurement data error is integrated out (11) or explicitly considered (12).

The second class of likelihood functions, 31-34, avoids over-conditioning of the likelihood surface in the presence of epistemic and other sources, and their mathematical formulation is guided by trial-and-error, expert knowledge, and commonly used goodness-of-fit criteria (Beven and Binley, 1992; Freer et al., 1996; Beven and Freer, 2001). These informal likelihood functions enable users to implement the GLUE methodology of Beven and Binley (1992). The use of DREAM enhances, sometimes dramatically, the computational efficiency

of GLUE (*Blasone et al.*, 2008).

The field thinning of DREAMPar allows the user to specify the thinning rate of each Markov chain to reduce memory requirements for high-dimensional target distributions. For instance, for a  $d = 100$  dimensional target distribution with  $N = 100$  and  $T = 10,000$ , one would need a staggering 100-million bytes of memory to store all the samples of the joint chains. Thinning applies to all the sampled chains, and stores only every  $K$ th visited state. This option reduces memory storage with a factor of  $T/K$ , and also decreases the autocorrelation between successively stored chain samples. A default value of  $K = 1$  (no thinning) is assumed in DREAM. Note, large values for  $K$  ( $K \gg 10$ ) can be rather wasteful as many visited states are not used in the computation of the posterior moments and/or plotting of the posterior parameter distributions.

Multi-chain methods can suffer convergence problems if one or more of the sampled chains have become stuck in a local area of attraction while in pursuit of the target distribution. This fallacy has been addressed in the function `check` of the C++ code and is used to detect and resolve aberrant trajectories. Dissident chains are more likely to appear if the target distribution is high-dimensional and the posterior response surface is non-smooth with many local optima and regions of attraction. These non-ideal properties are often the consequence of poor model numerics (*Clark and Kavetski*, 2010; *Schoups et al.*, 2010b) and hinder convergence of MCMC simulation methods to the target distribution. The field `outlier` of DREAMPar lists (in quotes) the name of the outlier detection test that is used to expose dissident chains. Options available to the user include the 'iqr' (*Upton and Cook*, 1996), 'grubbs' (*Grubbs*, 1950), 'peirce' (*Peirce*, 1852), and 'chauvenet' (*Chauvenet*, 1960) method. These nonparametric methods diagnose dissident chains by comparing the mean log-density values of each of the  $N$  sampled trajectories. The premise of this comparison is that the states visited by an outlier chain should have a much lower average density than their counterparts sampling the target distribution. Those chains diagnosed as outlier will give up their present position in the parameter space in lieu of the state of one of the other  $N - 1$  chains, chosen at random. This correction step violates detailed balance (irreversible transition) but is necessary in some cases to reach formally convergence to a limiting distribution. Numerical experiments have shown that the default option (`outlier = 'iqr'`) works well in practice. Note, the problem of outlier chains would be resolved if proposals are created from past states of the chains as used in DREAM<sub>(ZS)</sub>, DREAM<sub>(DZS)</sub> and MT-DREAM<sub>(ZS)</sub>. Dissident chains can then sample their own position and jump directly to the mode of the target if  $\gamma = 1$  (*ter Braak and Vrugt*, 2008; *Laloy and Vrugt*, 2012a). We will revisit this issue in section 7 of this paper.

The field `adapt_pCR` of `DREAMPar` defines whether the crossover probabilities,  $\mathbf{p}_{cr}$  are adaptively tuned during a DREAM run so as to maximize the normalized Euclidean distance between two successive chain states. The default setting of 'yes', can be set to 'no' and thus switched off by the user in the GUI. The selection probabilities are tuned only during burn-in of the chains to not destroy reversibility of the sampled chains.

The default choice of the jump rate in DREAM is derived from the value of  $s_d = 2.38^2/d$  in the RWM algorithm. This setting should lead to optimal acceptance rates for Gaussian and Student target distributions, but might not yield adequate acceptance rates for real-world studies involving complex multivariate posterior parameter distributions. The field `beta0` of `DREAMPar` allows the user to increase (decrease) the value of the jump rate,  $\gamma = 2.38\beta_0/\sqrt{2\delta d^*}$ , thereby improving the mixing of the individual chains. This  $\beta_0$ -correction is applied to all sampled proposals, with the exception of the unit jump rate used for mode jumping. Values of  $\beta_0 \in [1/4, 1/2]$  have shown to work well for parameter-rich groundwater and geophysical models (e.g. *Laloy et al. (2015)*).

#### 4.3. Input argument 3: `Par_info`

The variable `Par_info` stores all necessary information about the parameters of the target distribution, for instance their prior uncertainty ranges (for bounded search problems), starting values (initial state of each Markov chain), prior distribution (defines Metropolis acceptance probability) and boundary handling (what to do if out of feasible space), respectively. Table 3 lists the different fields of `Par_info` and summarizes their content, default values and variable types.

Table 3 DREAM input argument `Par_info`: Different fields, their default settings and variable types. These variables appear under the "Parameter space" or "Prior distribution" tab of the GUI.

| Field <code>Par_info</code> | Description         | Options                            | Default     | Type                 |
|-----------------------------|---------------------|------------------------------------|-------------|----------------------|
| <code>initial</code>        | Initial sample      | 'uniform'/'latin'/'normal'/'prior' |             | string               |
| <code>min</code>            | Minimum values      |                                    | $-\infty_d$ | $1 \times d$ -vector |
| <code>max</code>            | Maximum values      |                                    | $\infty_d$  | $1 \times d$ -vector |
| <code>boundhandling</code>  | Boundary handling   | 'reflect'/'bound'/'fold'/'none'    | 'none'      | string               |
| <code>mu</code>             | Mean 'normal'       |                                    |             | $1 \times d$ -vector |
| <code>cov</code>            | Covariance 'normal' |                                    |             | $d \times d$ -matrix |
| <code>prior</code>          | Prior distribution  |                                    |             |                      |

<sup>†</sup> Multiplicative case: Each cell of the  $d$ -array contains a different marginal prior pdf.

<sup>‡</sup> Multivariate case: An anonymous function with prior pdf is provided by user.

Most of the variables of `Par_info` can be set by the user in the GUI of DREAM Suite under the "Parameter space" tab.

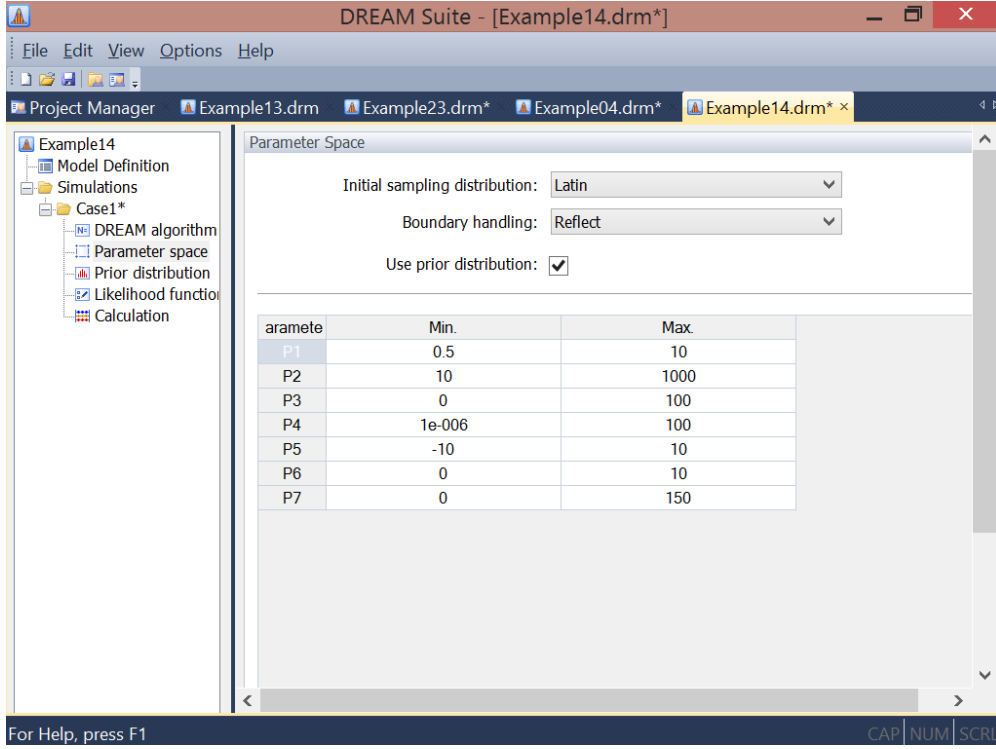


Figure 8 Screenshot of "Parameter space" tab of the GUI of DREAM Suite for example 14 of the project manager.

The field `initial` of `Par_info` specifies with a string enclosed between quotes how to sample the initial state of each of the  $N$  chains. Options available to the user include (1) 'uniform' (2) 'latin' (3) 'normal' and (4) 'prior', and they create the initial states of the chains by sampling from (1) a uniform prior distribution, (2) a Latin hypercube (*McKay et al.*, 1979), (3) a multivariate normal distribution, and (4) a user defined prior distribution. The first three options assume the prior distribution to be noninformative (uniform/flat), and consequently the posterior density of each proposal to be directly proportional to its likelihood. On the contrary, if the option 'prior' is used and a non-flat (informative) prior distribution of the parameters is specified by the user, then the density of each proposal becomes equivalent to the product of the (multiplicative) prior density and likelihood derived from the output of `model` plugin.

Option (1) and (2) require specification of the fields `min` and `max` of `Par_info`. These fields contain in a  $1 \times d$ -vector the lower and upper bound values of each of the parameters, respectively. If option (3) 'normal' is used then the fields `mu` ( $1 \times d$ -vector) and `cov` ( $d \times d$ -matrix) of `Par_info` should be defined by the user. These fields store the mean and covariance matrix of the multivariate normal distribution. We will revisit the option 'prior' at the end of this section.



The fields `min` and `max` of the `Par_info` serve two purposes. First, they define the feasible parameter space from which the initial state of each of the chains is drawn if 'uniform' random or 'latin' hypercube sampling is used. Second, they can define a bounded search domain for problems involving one or more parameters with known physical/conceptual ranges. This does however require the bound to be actively enforced during chain evolution. Indeed, proposals generated with Equations (19) and (20) can fall outside the hypercube defined by `min` and `max` even if the initial state of each chain are well within the feasible search space. The field `boundhandling` of `Par_info` provides several options what to do if the parameters are outside their respective ranges. The four different options that are available are (1) 'bound', (2) 'reflect', (3) 'fold', and (4) 'none' (default). These methods are illustrated graphically in Figure 9 and act on one parameter at a time.

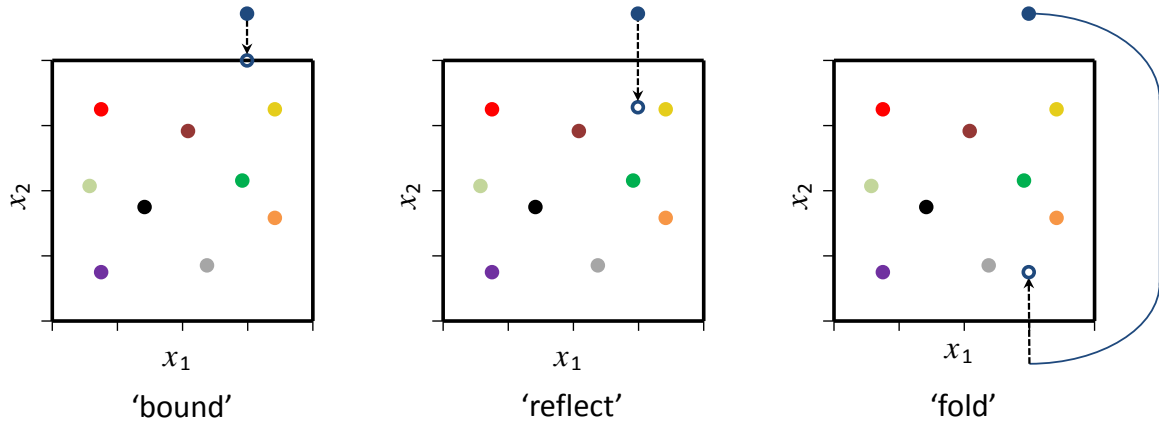


Figure 9 Different options for parameter treatment in bounded search spaces in the DREAM package. a) set to bound, b) reflection, and c) folding. The option folding is the only boundary handling approach that maintains detailed balance.

The option 'bound' is most simplistic and simply sets each parameter value that is out of bound of equal to its closest bound. The option 'reflect' is somewhat more refined and treats the boundary of the search space as a mirror through which each individual parameter value is reflected backwards into the search space. The reflection step size is simply equivalent to the "amount" of boundary violation. The 'bound' and 'reflect' options are used widely in the optimization literature in algorithms concerned only with finding the minimum (maximum, if appropriate) of a given cost or objective function. Unfortunately, these two proposal correction methods violate detailed balance in the context of MCMC simulation. It is easy to show for both boundary handling methods that the forward (correction step) and backward jump cannot be construed with equal probability. The third option 'fold' treats the parameter space as a continuum representation by simply connecting the upper bound of each dimension to its respective lower bound. This folding approach does not destroy the



Markovian properties of the  $N$  sampled chains, and is therefore preferred statistically. However, this approach can provide "bad" proposals (reduced acceptance rate) if the posterior distribution is located at the edges of the search domain. Then, the parameters can jump from one side of the search domain to the opposite end.

The option 'bound' is least recommended in practice as it collapses the parameter values to a single point. This not only relinquishes unnecessarily sample diversity but also inflates artificially the solution density (probability mass) at the bound. The loss of chain diversity also causes a-periodicity (proposal and current state are similar for selected dimensions) and distorts convergence to the target distribution. A simple numerical experiment with a truncated normal target distribution will demonstrate the superiority of the folding approach. This results in an exact inference of the target distribution whereas a reflection step overestimates the probability mass at the bound. For most practical applications, a reflection step will provide accurate results unless too many dimensions of the target distribution find their highest density in close vicinity of the bound.

What is left is a discussion of the use of 'prior' as initial sampling distribution of the chains. This option is specifically implemented to enable the use of an informative (non-flat) prior distribution. The user can select among two choices for the prior distribution, that is the use of a multiplicative prior or multivariate prior distribution. These options are available in the GUI on the "Prior distribution" tab and appear if the user selects the option "Use prior distribution" on the "Parameter space" tab.

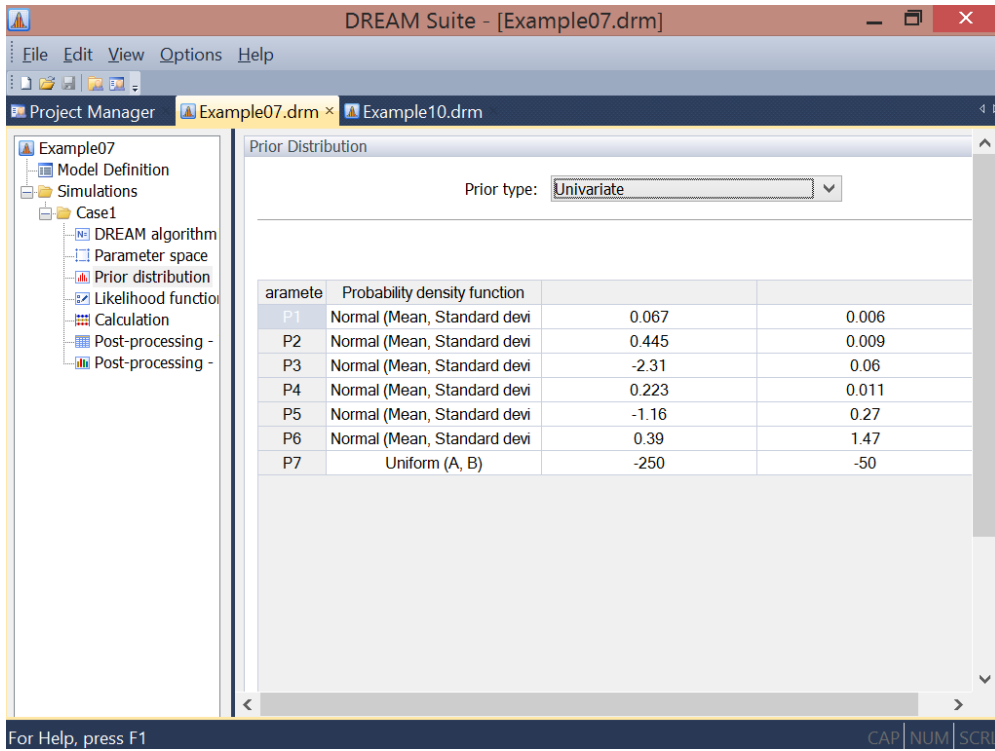


Figure 10 Screenshot of "Prior distribution" tab of the GUI of DREAM Suite for example 7 of the project manager.

In the multiplicative case each parameter has its own prior distribution. For instance, we can use a normal distribution with mean -2 and standard deviation of 4,  $\mathcal{N}(-2, 4)$  for the first parameter, a t-distribution with 10 degrees of freedom for the second parameter,  $\mathcal{T}(10)$ , and a uniform distribution between -10 and 20 for the third parameter,  $\mathcal{U}(-10, 20)$ . The GUI allows the user to define each of these distributions. The prior density of some parameter vector is then simply equivalent to the product of the individual pdfs specified by the user in the GUI. The user can select from the following list of built-in density functions in DREAM Suite: F, gamma, normal, T, and uniform.

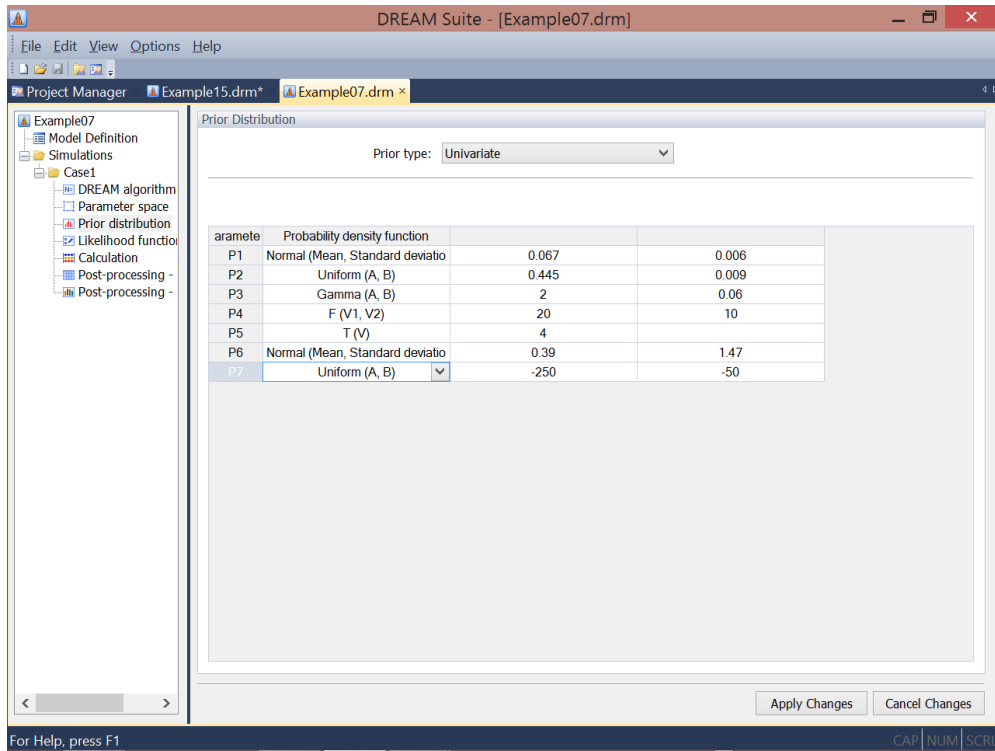


Figure 11 Screenshot of the "Prior distribution" tab of the GUI of DREAM Suite. This shot shows the different univariate prior distributions available to the user

The use of a multiplicative prior assumes the parameters of the prior distribution to be uncorrelated, an assumption that might not be justified for some inference problems. The GUI therefore also implements a second option where the user can select a multivariate prior distribution. Two options are available to the user including a  $d$ -variate normal distribution, and a  $d$ -variate Student t-distribution. For a multivariate normal distribution,  $\mathcal{N}_d(a, b)$  the user has to define the mean,  $a$  of the  $d$  parameter values and the  $d \times d$  covariance distribution,  $b$ . Alternatively, the user can implement a multivariate t-distribution,  $\mathcal{T}_d(a, b)$  with degrees of freedom  $a$  and correlation matrix,  $b$ . The values of  $a$  and  $b$  for both distributions can be entered in the GUI (see below).

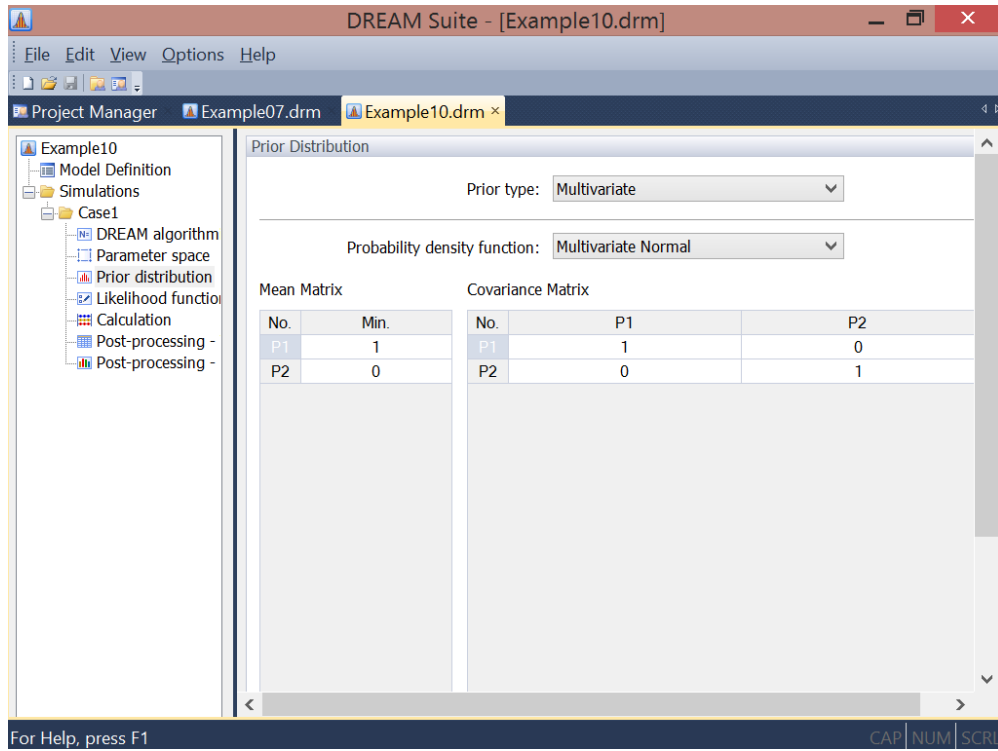


Figure 12 Screenshot of the "Prior distribution" tab of the GUI of DREAM Suite for example 10 of the program manager.

Note, the user can work with an informative prior distribution for each of the  $d$  parameters, and still sample the initial state of each Markov chain from the prior ranges of the parameters defined in the fields `min` and `max` of `Par_info`. This approach might be favored over sampling the initial states from the prior directly as it will allow DREAM to explore more thoroughly, at least in the first generations, the parameter space outside the prior pdf.

#### 4.4. (Optional) input argument 4: `Meas_info`

The fourth input argument `Meas_info` of the C++ code of DREAM is mandatory if the output of `model` constitutes a vector of simulated values or summary metrics of one or more entities of interest. Table 4 describes the different fields of `Meas_info`, their content and type. These fields and their values appear in DREAM Suite under the "Measurement data" tab of the GUI.

Table 4 Content of (optional) input **Meas\_info**. This fourth input argument of DREAM is required if the return argument of **model** constitutes a vector of simulated values (or summary statistics) of one or more variables. The field of **Meas\_info** appear under the "Measurement data" tab and "Summary statistics" tab of the GUI.

| Field <b>Meas_info</b> | Description              | Type                           | Tab of GUI           |
|------------------------|--------------------------|--------------------------------|----------------------|
| Y                      | Measurement data         | $n \times 1$ -vector           | "Measurement data"   |
| Sigma                  | Measurement error        | scalar or $n \times 1$ -vector | "Measurement data"   |
| S                      | Summary statistics (ABC) | $m \times 1$ -vector           | "Summary statistics" |

Two of the variables of **Meas\_info** can be defined by the user in the GUI of DREAM Suite under the "Measurement data" tab.

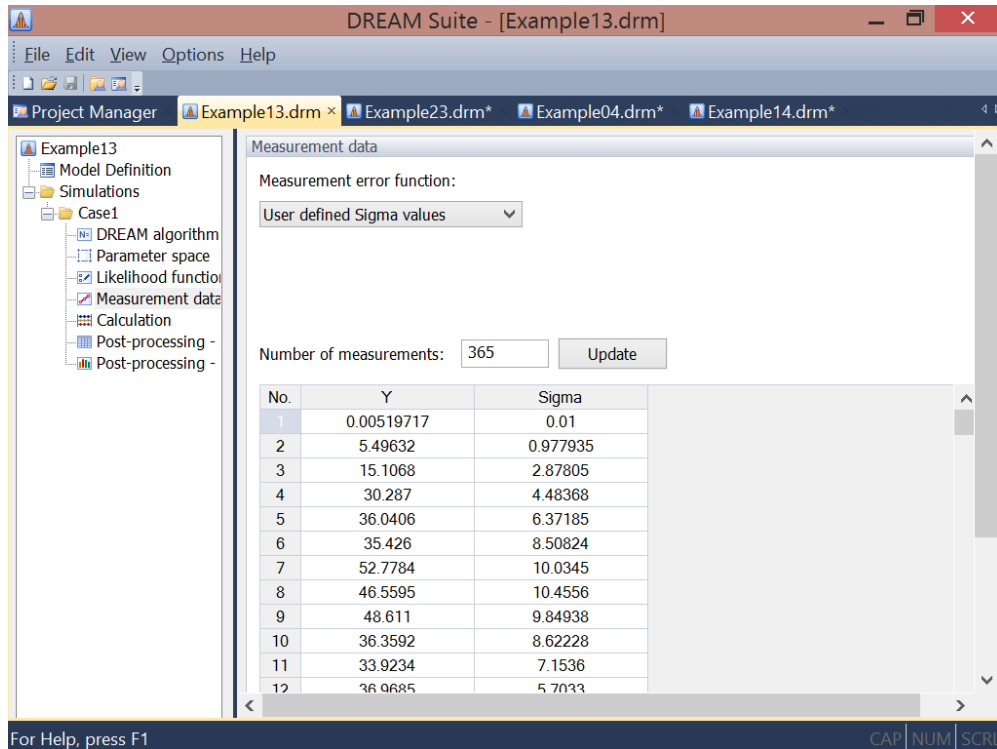


Figure 13 Screenshot of "Measurement data" tab of the GUI of DREAM Suite for example 13 of the project manager.

The field **Y** of **Meas\_info** stores the  $n \geq 1$  observations of the calibration data,  $\tilde{\mathbf{Y}}$  against which the output,  $\mathbf{Y}$  of **model** is compared. The  $n$ -vector of error residuals,  $\mathbf{E}(\mathbf{x}) = \tilde{\mathbf{Y}} - \mathbf{Y}(\mathbf{x})$  is then translated into a log-likelihood value using one of the formal (11-17) or informal (31-34) likelihood functions listed in Table 2 and defined by the user in field **lik** of **DREAMPar**.

The field **Sigma** of **Meas\_info** stores the measurement error of each entry of the field **Y**. This data error is necessary input for likelihood functions 12, 13 and 16. A single value for **Sigma** suffices if homoscedasticity of the data error is expected, otherwise  $n$ -values need to be declared and specify the heteroscedastic error of the observations of **Y**.

In case the measurement error of the data  $\mathbf{Y}$  is unknown, three different approaches can be implemented. The first option is to select likelihood function 11. This function is derived from Equation (7) by integrating over (out) the data measurement error. Field **Sigma** of **Meas\_info** can then be left blank (empty). The second option uses likelihood function 12, 13, or 16 and estimates the measurement data error along with the model parameters using nuisance variables. The user can then implement a homoscedastic or heteroscedastic measurement error model. Both appear under "Measurement error function" of the "Measurement data" tab and allow the user to implement a constant measurement error ("Homoscedastic error" in GUI), or **Sigma** = sig\_0, and a measurement error that increases with magnitude of the data ("Heteroscedastic error" in GUI), or **Sigma** = sig\_0 + sig\_1  $\times$   $\mathbf{Y}$ . The ranges of sig\_0 and sig\_1 can be defined by the user - yet their values should be strictly positive. If not, the measurement error can become smaller than zero. The nuisance variables sig\_0 and sig\_1 are jointly inferred with the parameters,  $\mathbf{x}$  of the model plugin defined by the user.

The third and last option that can be used if the measurement data error is unknown, uses likelihood function 14 (*Schoups and Vrugt, 2010a*) or 17 (*Scharnagl et al., 2015*). These functions do not use field **Sigma** (can be left empty) but rather use their own built-in measurement error model. The coefficients of the error models are part of a larger set of nuisance parameters that allow these likelihood functions to adapt to nontraditional error residual distributions. Appendix A summarizes the different likelihood functions and their nuisance variables.

The field **S** of **Meas\_info** stores  $m \geq 1$  summary statistics of the data, and is mandatory input for likelihood functions 21, 22, 23 used for ABC, diagnostic model evaluation, and limits of acceptability. The number of elements of  $\mathbf{Y}$  and **S** should match exactly the output of the script `model` written by the user. The summary statistics can be defined by the user in the "Summary statistics" tab in the GUI of DREAM Suite.

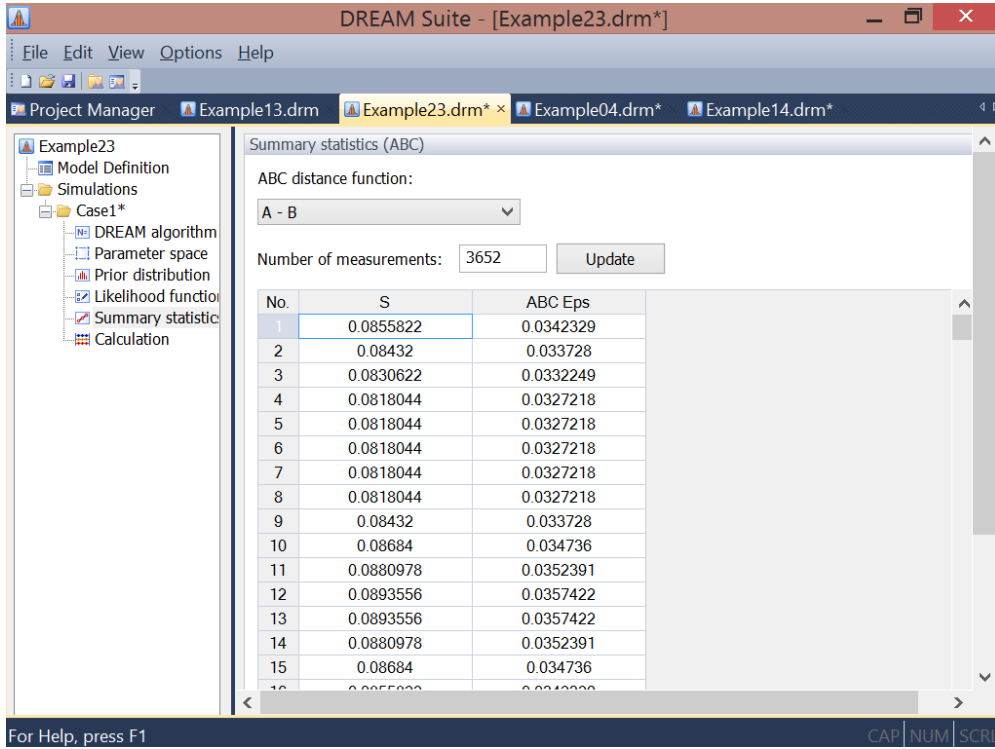


Figure 14 Screenshot of "Summary statistics" tab of the GUI of DREAM Suite for example 23 of the project manager.

#### 4.5. (Optional) input argument 5: options

The variable `options` is optional and passed as fifth input argument to DREAM. The fields of this variable can activate (among others) file writing, distributed multi-core calculation, storage of the model output simulations, ABC, diagnostic model evaluation, diagnostic Bayes, and the limits of acceptability framework. Table 5 summarizes the different fields of `options` and their default settings. The different variables that are stored in `options` appear in the GUI under different tabs. These tabs are listed in the last column.

Table 5 Content of (optional) input `options`. This fifth input argument of the main DREAM code is required to activate several of its built-in capabilities such as distributed multi-processor calculation, ABC, diagnostic model evaluation, diagnostic Bayes and limits of acceptability. The variables of input argument `options` appear under different tabs in the GUI, listed in the last column.

| Field options | Description                         | Tab of GUI            |
|---------------|-------------------------------------|-----------------------|
| parallel      | Distributed multi-core calculation? | "Calculation"         |
| DB            | Diagnostic Bayes?                   | "Likelihood function" |
| epsilon       | ABC cutoff threshold                | "Summary statistics " |
| rho           | ABC distance function               | "Summary statistics"  |

Multi-core calculation takes advantage of the Message Passing Interface (MPI) and evaluates the  $N$  different proposals created with Equations (19) and (20) on a different processor

and/or thread. Parallel computing is built-in DREAM Suite and automatically activated - unless the user deactivates this option in the "Calculation" tab of the GUI.

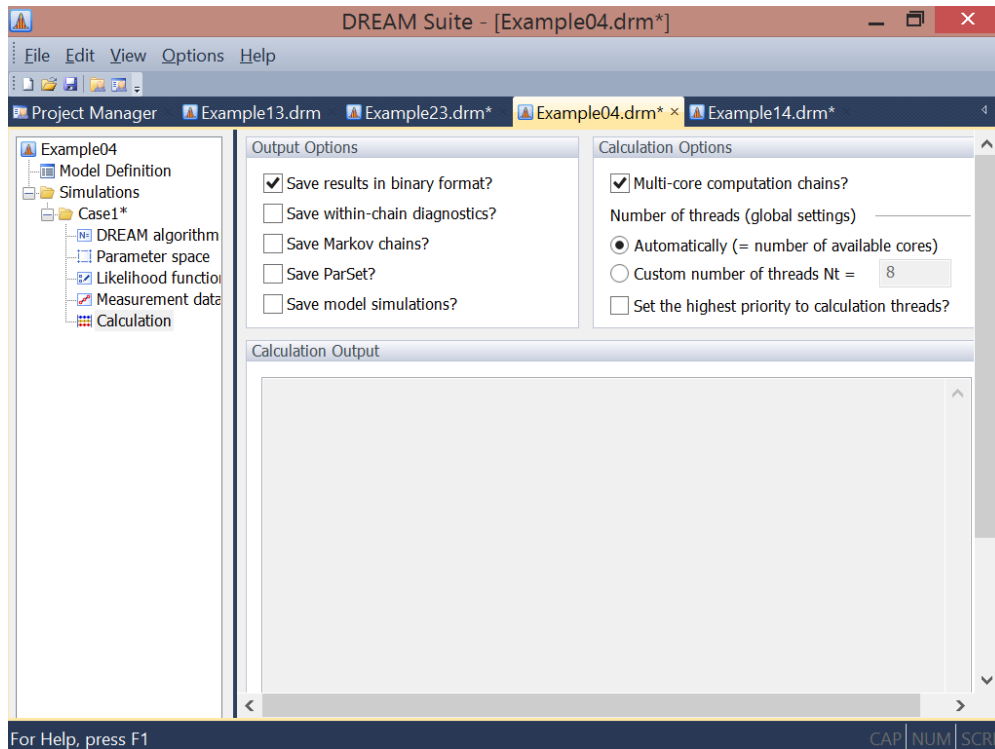


Figure 15 Screenshot of "Calculation" tab of the GUI of DREAM Suite for example 4 of the project manager.

Distributed calculation can significantly reduce the run time of DREAM Suite for CPU-demanding forward models. For simple models that require only a few seconds to run the time savings of a parallel run is usually negligible due to latency (transport delay) of the hardware and operating system.

For ABC or diagnostic model evaluation the fields **rho** and **epsilon** of variable **options** need to be specified unless their default settings are appropriate. The field **rho** specifies the mathematical formulation of the distance function between the simulated and observed summary statistics. In practice, a simple difference operator  $\rho = (A - B)$  (in GUI) between the observed and simulated observed summary statistics suffices. The field **epsilon** of **options** stores a small positive value which is used to truncate the behavioral (posterior) parameter space.

If ABC is used then the user can select two different implementations to solve for the target distribution. The first approach, adopted from *Turner and Sederberg (2012)*, uses likelihood function 21 to transform the distance function between the observed and simulated summary metrics in a probability density that DREAM uses to derive the target distribution.



This approach can produce nicely bell-shaped marginal distributions, but does not guarantee that the posterior summary metrics fall within **epsilon** of their observed values. A more viable and powerful approach was introduced recently by *Sadegh and Vrugt* (2014) and uses likelihood function 22 with the following modified Metropolis acceptance probability to decide whether to accept proposals or not

$$p_{\text{acc}}(\mathbf{X}^i \rightarrow \mathbf{X}_p^i) = \begin{cases} I(f(\mathbf{X}_p^i) \geq f(\mathbf{X}^i)) & \text{if } f(\mathbf{X}_p^i) < 0 \\ 1 & \text{if } f(\mathbf{X}_p^i) \geq 0 \end{cases}, \quad (22)$$

where  $I(a)$  is an indicator function that returns one if  $a$  is true, and zero otherwise. The mathematical expression of the fitness (likelihood) function 22 is given in Table ?? (in Appendix A). Equation (22) is implemented in an extension of DREAM called DREAM<sub>(ABC)</sub> and rapidly guides the posterior summary metrics to lie within **epsilon** of their observed counterparts.

#### 4.6. Convergence diagnostics & burn-in

From MCMC theory, the chains are expected to eventually converge to a stationary distribution, which should be the desired target distribution. But, how do we actually assess that convergence has been achieved in practice, without knowledge of the actual target distribution?

One way to check for convergence is to see how well the chains are mixing, or moving around the parameter space. For a properly converged MCMC sampler, the chains should sample, for a sufficiently long period, the approximate same part of the parameter space, and mingle readily and in harmony with one another around some fixed mean value. This can be inspected visually for each dimension of  $\mathbf{x}$  separately, and used to diagnose convergence informally.

Another proxy for convergence monitoring is the acceptance rate. A value between 15 - 30% is usually indicative of good performance of a MCMC simulation method. Much lower values usually convey that the posterior surface is difficult to traverse in pursuit of the target distribution. A low acceptance rate can have different reasons, for instance poor model numerics, or the presence of multi-modality and local optima. The user can enhance the acceptance rate by declaring a value for  $\beta_0 < 1$  in field **beta0** of structure **DREAMP**ar (see Table 1). This multiplier will reduce the jumping distance,  $d\mathbf{X}$  in Equation (19) and thus proposals will remain in closer vicinity of the current state of each chain. This should enhance the acceptance rate and mixing of individual chains. Note, the acceptance rate can only diagnose whether a MCMC method such as DREAM is achieving an acceptable performance, it cannot be used to determine when convergence has been achieved.

The C++ code of DREAM includes various non-parametric and parametric statistical tests to determine when convergence of the sampled chains to a limiting distribution has been achieved. The most powerful of these convergence tests is the univariate,  $\hat{R}$ -statistic of *Gelman and Rubin* (1992) and the multivariate  $\hat{R}^d$  statistic of *Brooks and Gelman* (1998). The univariate  $\hat{R}$ -statistic compares for each parameter  $j = \{1, \dots, d\}$  the within-chain

$$W_j = \frac{2}{N(T-2)} \sum_{r=1}^N \sum_{i=\lfloor T/2 \rfloor}^T (\mathbf{x}_{i,j}^r - \bar{\mathbf{x}}_j^r)^2 \quad \bar{\mathbf{x}}_j^r = \frac{2}{T-2} \sum_{i=\lfloor T/2 \rfloor}^T \mathbf{x}_{i,j}^r \quad (23)$$

and between-chain variance

$$B_j/T = \frac{1}{2(N-1)} \sum_{r=1}^N (\bar{\mathbf{x}}_j^r - \bar{\bar{\mathbf{x}}}_j)^2 \quad \bar{\bar{\mathbf{x}}}_j = \frac{1}{N} \sum_{r=1}^N \bar{\mathbf{x}}_j^r \quad (24)$$

using

$$\hat{R}_j = \sqrt{\frac{N+1}{N} \frac{\hat{\sigma}_+^{2(j)}}{W_j} - \frac{T-2}{NT}}, \quad (25)$$

where  $T$  signifies the number of samples in each chain,  $\lfloor \cdot \rfloor$  is the integer rounding operator, and  $\hat{\sigma}_+^{2(j)}$  is an estimate of the variance of the  $j$ th parameter of the target distribution

$$\hat{\sigma}_+^{2(j)} = \frac{T-2}{T} W_j + \frac{2}{T} B_j. \quad (26)$$

To official declare convergence, the value  $\hat{R}_j \leq 1.2$  for each parameter,  $j \in \{1, \dots, d\}$ , otherwise the value of  $T$  should be increased and the chains run longer. As the  $N$  different chains are launched from different starting points, the  $\hat{R}$ -diagnostic is a relatively robust estimator.

The C++ code also computes the multivariate  $\hat{R}$ -statistic, also referred to  $\hat{R}^d$ -diagnostic. This metric is defined in *Brooks and Gelman* (1998) and assesses convergence of the  $d$  parameters simultaneously by comparing their within and between-sequence covariance matrix. Convergence is achieved when a rotationally invariant distance measure between the two matrices indicates that they are "sufficiently" close. Then, the multivariate  $\hat{R}^d$ -statistic achieves a value close to unity, otherwise its value is larger than one. In fact, the  $\hat{R}$  and  $\hat{R}^d$ -statistic take on a very similar range of values, hence simplifying analysis of when convergence has been achieved. Practically experience suggests that the  $\hat{R}^d$ -statistic is particularly useful for high-dimensional target distributions involving complicated multi-dimensional parameter interactions.

The traceplots of the  $\hat{R}$  and  $\hat{R}^d$ -convergence diagnostics appear in the postprocessor of

DREAM Suite (see next section) and can be used to determine when convergence has been achieved and thus which samples of `chain` to use for posterior estimation and analysis. The other samples can simply be discarded from the chains as burn-in.

The C++ code of DREAM also computes several within-chain diagnostics. Their values are not listed by the postprocessor, but instead saves to a binary or text file if activated by the user in the "Calculation" tab of the GUI. The within-chain diagnostics file contains several convergence diagnostics including, the autocorrelation function, the *Geweke* (1992), and *Raftery and Lewis* (1992)-diagnostics. These three within-chain diagnostics are calculated for each of the  $N$  chains and  $d$  parameters separately.

The autocorrelation function for each parameter  $j = \{1, \dots, d\}$  is defined as

$$\rho_{j,k}^r = \frac{\sum_{i=1}^{T-k} (\mathbf{x}_{i,j}^r - \bar{\mathbf{x}}_j^r)(\mathbf{x}_{i+k,j}^r - \bar{\mathbf{x}}_j^r)}{\sum_{i=1}^T (\mathbf{x}_{i,j}^r - \bar{\mathbf{x}}_j^r)^2}, \quad (27)$$

and returns the correlation between two samples  $k$  iterations apart in the  $r$ th chain,  $r = \{1, \dots, N\}$ . Compared to rejection sampling which, per construction, produces uncorrelated samples, MCMC chain trajectories exhibit autocorrelation as the current state of the chain is derived from its previous state. This correlation is expected to decrease with increasing lag  $k$ . The autocorrelation function is a useful proxy to assess sample variability and mixing, but does not convey when convergence has been achieved. A high autocorrelation, say  $|\rho| > 0.8$ , at lags, say  $k \geq 5$ , simply demonstrates a rather poor mixing of the individual chains.

The *Geweke* (1992)-diagnostic compares the means of two nonoverlapping parts of the Markov chain using a standard Z-score adjusted for autocorrelation. The *Raftery and Lewis* (1992)-statistic calculates the number of iterations,  $T$  and length of burn-in necessary to satisfy the condition that some posterior quantile of interest, say  $q$  has a probability,  $p$  of lying within interval  $[q - r, q + r]$ . Default values are  $q = 0.025$ ,  $p = 0.95$ , and  $r = 0.01$ , respectively. Details of how to compute and interpret these two statistics is found in the cited references.

Altogether, joint interpretation of the different diagnostics should help assess convergence of the sampled chain trajectories. Of all these metrics, the  $\hat{R}$ -statistic provides the best guidance on exactly when convergence has been achieved. This happens as soon as this statistic drops below the critical threshold of 1.2 for all  $d$  parameters of the target distribution. Suppose this happens at  $T^*$  iterations (generations) then the first  $(T^* - 1)$  samples of each chain are simply discarded as burn-in and the remaining  $N(T - T^*)$  samples from the joint chains are used for posterior analysis. Note, I always recommend to verify convergence of DREAM by visually inspecting the mixing of the different chain trajectories.

In practice, one has to make sure that a sufficient number of chain samples is available for the inference, otherwise the posterior estimates can be biased. For convenience, I list here the total number of posterior samples,  $N(T - T^*)$  (in brackets) one would need for a reliable inference with DREAM for a given dimensionality of the target distribution:  $d = 1$  (500);  $d = 2$  (1,000);  $d = 5$  (5,000);  $d = 10$  (10,000);  $d = 25$  (50,000);  $d = 50$  (200,000);  $d = 100$  (1,000,000);  $d = 250$  (5,000,000). These listed numbers are only a rough guideline, and based on several assumptions such as a reasonable acceptance rate ( $> 10\%$ ) and not too complicated shape of the posterior distribution. In general, the number of posterior samples required increases with rejection rate and complexity of the target distribution.

#### 4.7. Output arguments

I now briefly discuss the three output (return) arguments of DREAM including **chain**, **output** and **fx**. These three variables summarize the results of the DREAM algorithm and are used for convergence assessment, posterior analysis and plotting.

The variable **chain** is a matrix of size  $T \times d + 2 \times N$ . The first  $d$  columns of **chain** store the sampled parameter values (state), whereas the subsequent two columns list the associated log-prior and log-likelihood values respectively. If thinning is applied to each of the Markov chains then the number of rows of **chain** is equivalent to  $T/K + 1$ , where  $K \geq 2$  denotes the thinning rate. If a non-informative (uniform) prior is used then the values in column  $d + 1$  of **chain** are all zero and consequently,  $p(\mathbf{x}|\tilde{\mathbf{Y}}) \propto L(\mathbf{x}|\tilde{\mathbf{Y}})$ . With an informative prior, the values in column  $d + 1$  are non-zero and the posterior density,  $p(\mathbf{x}|\tilde{\mathbf{Y}}) \propto p(\mathbf{x})L(\mathbf{x}|\tilde{\mathbf{Y}})$ .

The content of **chain** can be used to calculate the posterior statistics, such as the maximum a-posteriori density (MAP) solution, the posterior mean, the posterior median, and the posterior standard deviation.

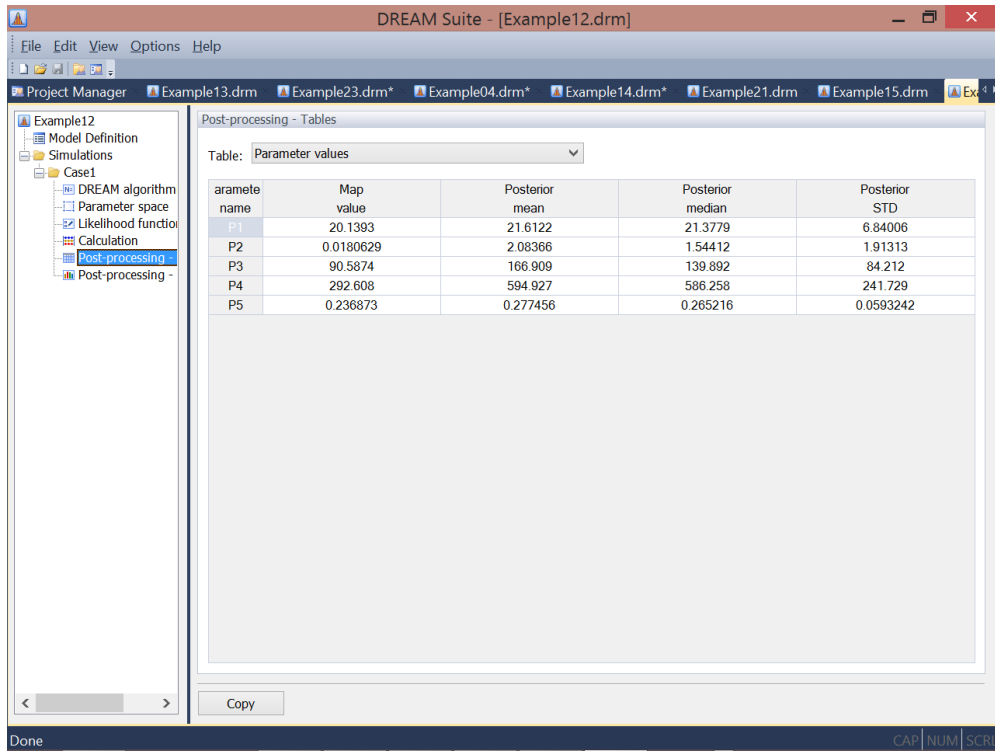


Figure 16 Screenshot of "Postprocessing-Tables" tab of the GUI of DREAM Suite for example 12 of the project manager.

The variable `output` contains important (diagnostic) information about the progress of the DREAM algorithm. The field `RunTime` (scalar) stores the wall-time (seconds), `R_stat` (matrix), `MR_stat`, `AR` (matrix) and `CR` (matrix) list for a given number of generations the  $\hat{R}$  convergence diagnostic for each individual parameter of the target distribution, the  $\hat{R}^d$  convergence diagnostic as function of number of generations, the average acceptance rate, and the selection probability of each of the  $n_{CR}$  crossover values, respectively. All the variables provide insights about the performance and convergence rate of the DREAM algorithm.

The  $\hat{R}$  and  $\hat{R}^d$ -statistics appear in the "Postprocessing-Charts" tab of the DREAM Suite GUI.

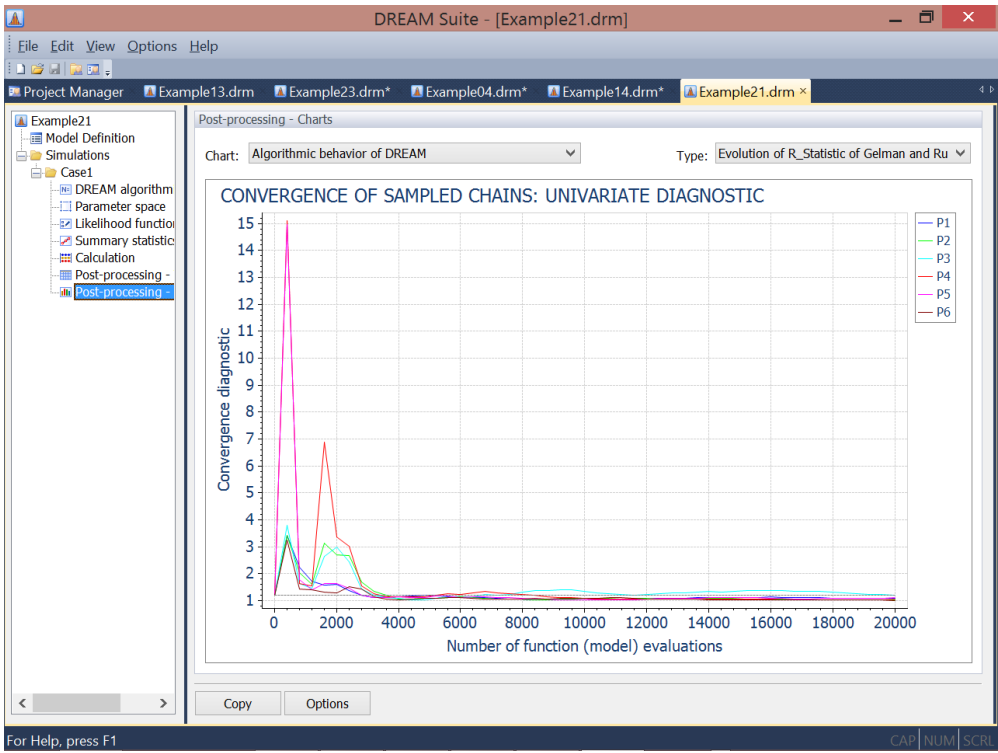


Figure 17 Screenshot of "Postprocessing-Charts" tab of the GUI of DREAM Suite for example 21 of the project manager.

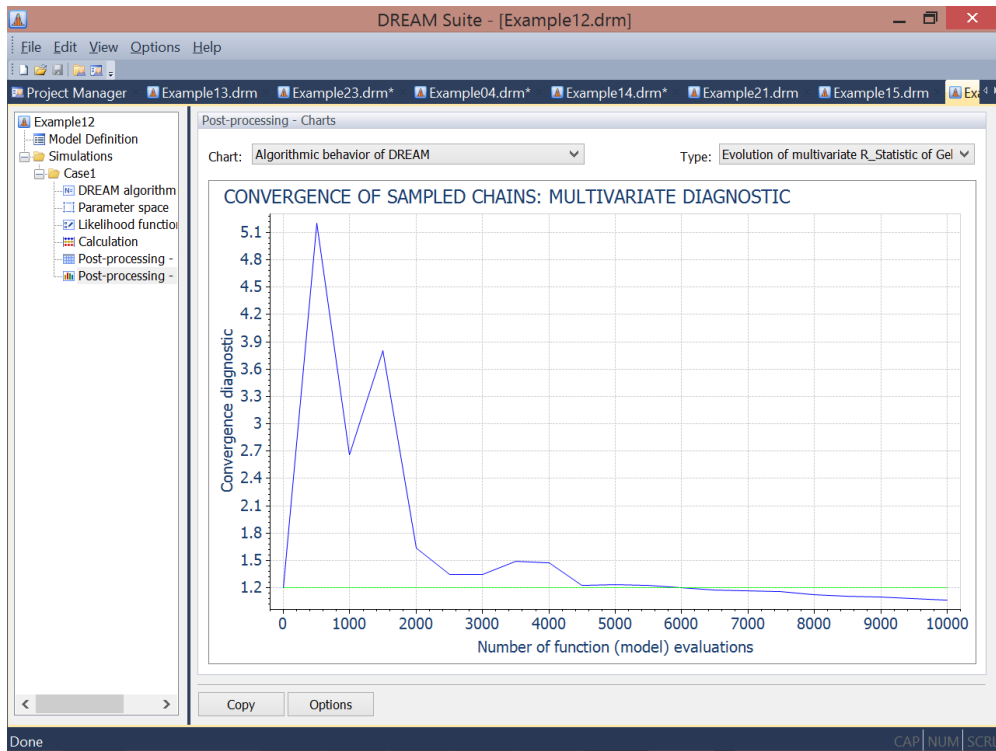


Figure 18 Screenshot of "Postprocessing-Charts" tab of the GUI of DREAM Suite for example 12 of the project manager.

The acceptance rate is also plotted in the by the postprocessor in the GUI.

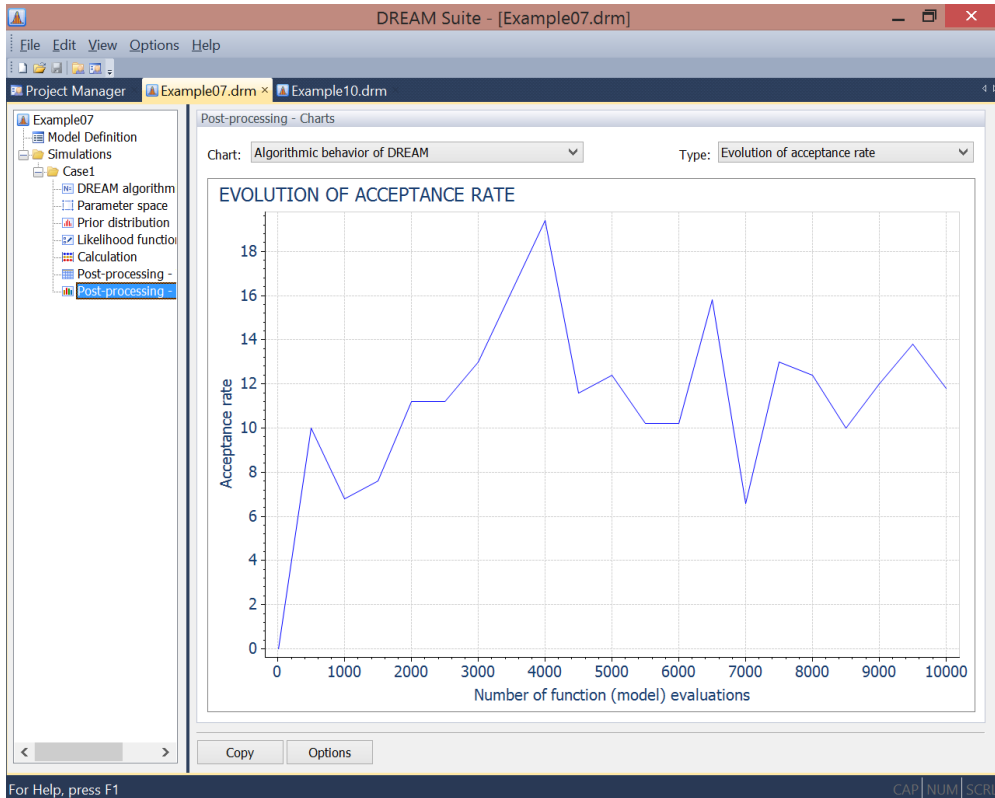


Figure 19 Screenshot of "Postprocessing-Charts" tab of the GUI of DREAM Suite for example 7 of the project manager.

Finally, the matrix  $\mathbf{fx}$  stores the output  $\mathbf{Y}$  of `model`. If this return argument constitutes a vector of simulated values (summary metrics) then  $\mathbf{fx}$  is of size  $NT \times n$  ( $NT \times m$ ), otherwise  $\mathbf{fx}$  is a vector of  $NT \times 1$  with likelihood or log-likelihood values. If thinning is used then this applies to  $\mathbf{fx}$  as well and the number of rows of  $\mathbf{fx}$  becomes equivalent to  $NT/K + 1$ ;  $K \geq 2$ .

All the output of the DREAM algorithm is visualized by the postprocessor of DREAM Suite. The output data can also be saved to different binary or text files using the "Calculation" tab of the GUI (see below).



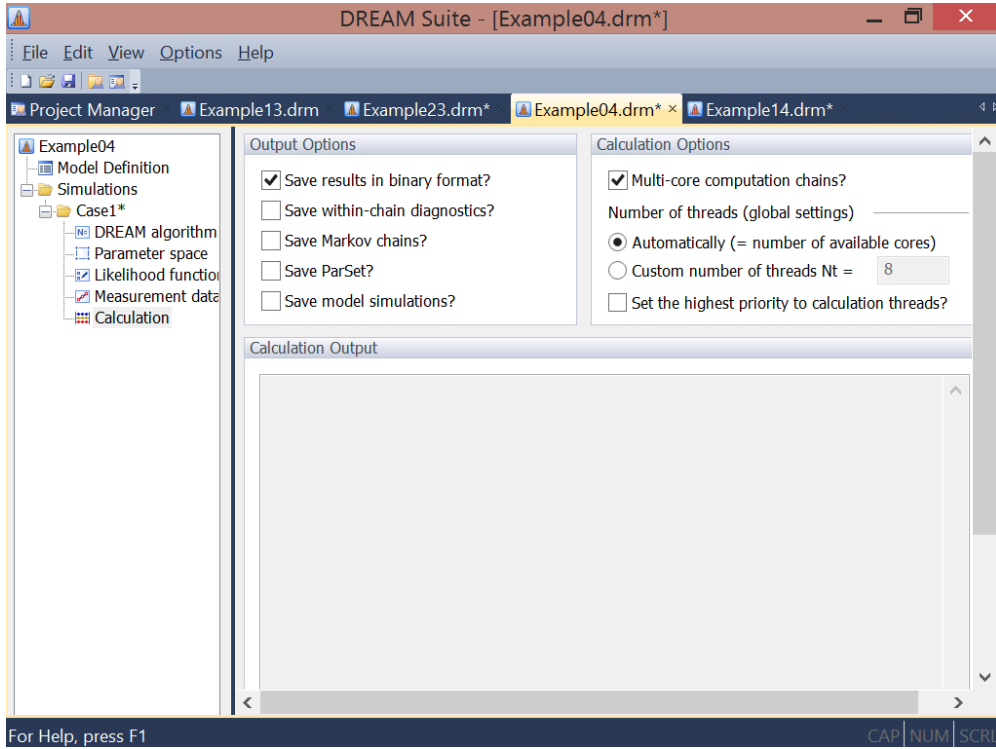


Figure 20 Screenshot of "Calculation" tab of the GUI of DREAM Suite for example 4 of the project manager.

## 5. Miscellaneous

In this section I discuss a few methods that can be used with DREAM Suite.

## 6. Diagnostic Bayes

A recurrent issue with the application of ABC is self-sufficiency of the summary metrics,  $S(\tilde{\mathbf{Y}})$ . In theory,  $S(\cdot)$  should contain as much information as the original data itself, yet complex systems rarely admit sufficient statistics. *Vrugt* (2015b) therefore proposed in another recent article a hybrid approach, coined diagnostic Bayes, that uses the summary metrics as prior distribution and the original data in the likelihood function, or  $p(\mathbf{x}|\tilde{\mathbf{Y}}) \propto p(\mathbf{x}|S(\tilde{\mathbf{Y}}))L(\mathbf{x}|\tilde{\mathbf{Y}})$ . This approach guarantees that no information is lost during the inference. The use of summary metrics as prior distribution is rather unorthodox and arguments for this approach are given by *Vrugt* (2015b).

Diagnostic Bayes is easily setup and executed within DREAM Suite. The user has to select the option "Diagnostic Bayes" under the "Likelihood function" tab of the GUI. The observations of the calibration data are stored under the tab "Measurement data" and related summary statistics are stored in the tab "Summary statistics".

### 6.1. Joint Parameter and State Estimation

The return argument  $\mathbf{Y}$  of the function script `model` usually involves the output of some model,  $\mathbf{Y} \leftarrow \mathcal{F}(\mathbf{x}, \cdot)$ . The computation in this script can involve state estimation as well. The return argument of `model` then involves a time-series of forecasts derived from the Kalman filter. This approach, assumes time-invariant parameter values and is at the heart of SODA and particle-DREAM (*Vrugt et al.*, 2005, 2013b).

### 6.2. $\ell^\Omega$ -Norm of Error Residuals

Likelihood functions play a key role in statistical inference of the model parameters. Their mathematical formulation depends on the assumptions that are made about the probabilistic properties of the error residuals. The validity of these assumptions can be verified a-posteriori by inspecting the actual error residual time series of the posterior mean simulation. Likelihood functions based on a  $\ell^2$ -norm (squared residuals) are most often used in practical applications, despite their relative sensitivity to peaks and outlier data points. Their use is motivated by analytic tractability - that is - with relatively little ease confidence intervals of the parameters can be construed from a classical first-order approximation around the optimum. This attractive feature of a  $\ell^2$ -type likelihood function was of imminent importance in past eras without adequate computational resources but is a far less desirable quality nowadays with availability of powerful computing capabilities and efficient algorithms. Indeed, methods such as DREAM can solve for likelihood functions with any desired norm,  $\Omega \in \mathbb{N}^+$ . For instance, the Laplacian likelihood (see Table ??) uses a  $\ell^1$  norm of the error residuals and therefore should be less sensitive to peaks and outliers. Unless there are very good reasons to adopt a  $\ell^2$ -type likelihood function, their use might otherwise be a historical relic (*Beven and Binley*, 2014).

## 7. THE DREAM FAMILY OF ALGORITHMS

In the past years, several other MCMC algorithms have appeared in the literature with a high DREAM pedigree. These algorithms use DREAM as their basic building block but include special extensions to simplify inference (among others) of discrete and combinatorial search spaces, and high-dimensional and CPU-intensive system models. These algorithms are described in detail in the literature. I refer the reader to *Vrugt* (2016) for a summary of these methods. We are planning to include these methods in the C++ code in the near future.

## 8. SUMMARY

In this manual I have reviewed the basic theory of Markov chain Monte Carlo (MCMC) simulation, and have introduced the basic elements of the C++ code of DREAM. The input variables of DREAM have been discussed and their implementation in the GUI of DREAM Suite. The DREAM algorithm provides scientists and engineers with an arsenal of options and utilities to solve posterior sampling problems involving (amongst others) bimodality, high-dimensionality, summary statistics, bounded parameter spaces, dynamic simulation models, formal/informal likelihood functions, diagnostic model evaluation, data assimilation, Bayesian model averaging, distributed computation, and informative/noninformative prior distributions. DREAM Suite supports parallel computing and includes tools for convergence analysis of the sampled chain trajectories and post-processing of the results. The output of DREAM is visualized in tables and figures by the postprocessor of DREAM Suite.

## Appendix A. Mathematical formulation of built-in likelihood functions

The mathematical formulations of the built-in likelihood functions of DREAM in Table 2 are given in Table A.1 below. For convenience,  $\mathbf{E}(\mathbf{x}) = \{e_1(\mathbf{x}), \dots, e_n(\mathbf{x})\}$  signifies the  $n$ -vector of residuals,  $\tilde{\mathbf{S}} = \{S_1(\tilde{\mathbf{Y}}), \dots, S_m(\tilde{\mathbf{Y}})\}$  and  $\mathbf{S} = \{S_1(\mathbf{Y}(\mathbf{x})), \dots, S_m(\mathbf{Y}(\mathbf{x}))\}$  are  $m$ -vectors with observed and simulated summary statistics, respectively, and  $\mathbf{A} = \{a_1, \dots, a_n\}$  is a  $n$ -vector of filtered residuals in likelihood function 14 using an autoregressive model with coefficients,  $\boldsymbol{\phi} = \{\phi_1, \dots, \phi_4\}$ .

Table A.1 Mathematical formulation of built-in likelihood functions of DREAM Suite. Option (1) and (2) return directly a likelihood and log-likelihood value, respectively, and their formulation is defined in the `model` script by the user.

| lik  | Mathematical formulation  | Latent variables  | Note |
|--|---|---|------|
| Formal likelihood functions                            |   |   |      |
| 11 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $-\frac{n}{2} \log \left\{ \sum_{t=1}^n e_t(\mathbf{x})^2 \right\}$   | none  |      |
| 12 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $-\frac{n}{2} \log(2\pi) - \sum_{t=1}^n \{ \log(\sigma_t) \} - \frac{1}{2} \sum_{t=1}^n \left( \frac{e_t(\mathbf{x})}{\sigma_t} \right)^2$  | $\sigma_t; t \in \{1, \dots, n\}$   | †    |
| 13 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $-\frac{n}{2} \log(2\pi) - \frac{1}{2} \log \left( \frac{\sigma_1^2}{(1-\phi^2)} \right) - \frac{1}{2} (1-\phi^2) \left( \frac{e_1(\mathbf{x})}{\sigma_1} \right)^2$<br>$-\sum_{t=2}^n \{ \log(\sigma_t) \} - \frac{1}{2} \sum_{t=2}^n \left( \frac{(e_t(\mathbf{x}) - \phi e_{t-1}(\mathbf{x}))}{\sigma_t} \right)^2$  | $\sigma_t, \phi; t \in \{1, \dots, n\}$   | †    |
| 14 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) \simeq$ | $n \log \left( \omega_\beta \frac{2\sigma_\xi}{(\xi + \xi^{-1})} \right) - \sum_{t=1}^n \{ \log(\sigma_t) \} - c_\beta \sum_{t=1}^n  a_{\xi,t} ^{2/(1+\beta)}$<br>$+ (\lambda_{\text{BC}} - 1) \sum_{t=1}^n (\tilde{y}_t + K_{\text{BC}})$  | $\sigma_0, \sigma_1, \beta, \xi, \mu_1, \phi, K_{\text{BC}}, \lambda_{\text{BC}}$ | ‡§   |
| 15 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $\sum_{j=1}^{\lfloor n/2 \rfloor} \left\{ \log(f_{\mathcal{F}}(\lambda_j, \mathbf{x}) + f_{\mathbf{E}}(\lambda_j, \Phi)) + \frac{g(\lambda_j)}{f_{\mathcal{F}}(\lambda_j, \mathbf{x}) + f_{\mathbf{E}}(\lambda_j, \Phi)} \right\}$  | none  | ¶    |
| 16 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $-\sum_{t=1}^n \{ \log(2\sigma_t) \} - \sum_{t=1}^n \left( \frac{ e_t(\mathbf{x}) }{\sigma_t} \right)$  | $\sigma_t; t \in \{1, \dots, n\}$   | †    |
| 17 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $\sum_{t=2}^n \left\{ \log \left( 2c_2 \Gamma((\nu+1)/2) \sqrt{\nu/(\nu-2)} \right) \right.$<br>$\left. - \log \left( (\kappa + \kappa^{-1}) \Gamma(\nu/2) \sqrt{\pi\nu} \sqrt{(1-\phi^2)\sigma_t} \right) \right.$<br>$\left. - ((\nu+1)/2) \log \left( 1 + (1/(\nu-2)) \left( \frac{c_1 + c_2 \eta_t}{\kappa^{\text{sign}(c_1 + c_2 \eta_t)}} \right)^2 \right) \right\}$ | $\sigma_a, \sigma_b, \sigma_c, \sigma_d, \phi, \nu, \kappa$                       | §¥   |
| ABC - diagnostic model evaluation                      |   |   |      |
| 21 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $-\frac{m}{2} \log(2\pi) - m \log(\epsilon) - \frac{1}{2} \epsilon^{-2} \sum_{j=1}^m \rho(S_j(\tilde{\mathbf{Y}}), S_j(\mathbf{Y}(\mathbf{x})))^2$  | none  | ℒ‡   |
| 22 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $\min_{j=1:m} \left( \epsilon_j - \rho(S_j(\tilde{\mathbf{Y}}), S_j(\mathbf{Y}(\mathbf{x}))) \right)$   | none  | ℒ‡   |
| GLUE - limits of acceptability                         |   |   |      |
| 23 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $\sum_{j=1}^m \left\{ I( S_j(\tilde{\mathbf{Y}}) - S_j(\mathbf{Y}(\mathbf{x}))  \leq \epsilon_j) \right\}$  | none  | ♣‡   |
| GLUE - informal likelihood functions                   |   |   |      |
| 31 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $-G \log \{ \text{Var}[\mathbf{E}(\mathbf{x})] \}$  | none  | ◇    |
| 32 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $G \log \left( 1 - \frac{\text{Var}[\mathbf{E}(\mathbf{x})]}{\text{Var}[\tilde{\mathbf{Y}}]} \right)$   | none  | ◇    |
| 33 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $-G \text{Var}[\mathbf{E}(\mathbf{x})]$   | none  | ◇    |
| 34 $\mathcal{L}(\mathbf{x} \tilde{\mathbf{Y}}) =$      | $-\log \left\{ \sum_{t=1}^n  e_t(\mathbf{x})  \right\}$   | none  | ◇    |

† Measurement error,  $\sigma_t$  defined in field `Sigma` of `Meas_info` or inferred jointly with  $\mathbf{x}$

‡ Measurement error defined as  $\sigma_t = \sigma_0 + \sigma_1 y_t(\mathbf{x})$ ; Scalars  $\omega_\beta$ ,  $\sigma_\xi$  and  $c_\beta$  derived from values of  $\xi$  and  $\beta$ ;  $\phi = \{\phi_1, \dots, \phi_4\}$  stores coefficients autoregressive model of error residuals

§ User is free to select exact formulation (depends on selection nuisance variables)

¶ Fourier frequencies,  $\lambda_j$ , spectral density function,  $f_{\mathbf{E}}(\cdot)$  and periodogram,  $g(\cdot)$  defined in *Whittle* (1953)

¥ Scalars  $c_1$  and  $c_2$  computed from  $\nu > 2$  and  $\kappa > 0$ ;  $\eta$  signifies  $(n-1)$ -vector of restandardized first-order decorrelated residuals;  $\Gamma(\cdot)$  and  $\text{sign}$  denote the gamma and signum function, respectively

ℒ ABC distance function,  $\rho(S(\tilde{\mathbf{Y}}), S(\mathbf{Y}(\mathbf{x})))$  specified as field `rho` of `options`

‡  $\epsilon$  (scalar or  $m$ -vector) stored in field `epsilon` of `options`

♣ Variable  $I(a)$  returns one if  $a$  is true, zero otherwise

◇ Shaping factor,  $G$  defined in field `GLUE` of `DREAMPar`. Default setting of  $G = 10$

The generalized likelihood function of *Schoups and Vrugt* (2010a) allows for bias correction, which is applied to the first or higher order filtered residuals prior to calculation of the likelihood. I refer to *Schoups and Vrugt* (2010a) and *Scharnagl et al.* (2015) for an exact derivation and detailed analysis of likelihood functions 14 and 17, respectively, and *Whittle* (1953) for an introduction to likelihood 15. The ABC likelihood functions 21 and 22 are described and discussed in detail by *Turner and Sederberg* (2012) and *Sadegh and Vrugt* (2014), whereas the limits of acceptability function 23 is introduced and tested in *Vrugt* (2015a). The pseudo-likelihoods in 31, 32, 33 and 34 are explicated in the GLUE papers of *Beven* and coworkers (*Beven and Binley*, 1992; *Freer et al.*, 1996; *Beven and Freer*, 2001; *Beven*, 2006). The derivation and explanation of the remaining likelihood functions, 11, 12, 13, and 16 can be found in introductory textbooks on time-series analysis and Bayesian inference.

Likelihood functions 14 and 17 extend the applicability of the other likelihood functions to situations where residual errors are correlated, heteroscedastic, and non-Gaussian with varying degrees of kurtosis and skewness. For instance, consider Figure A.1 which plots the density of the generalized likelihood function for different values of the skewness,  $\beta$  and kurtosis,  $\xi$ . The density is symmetric for  $\xi = 1$ , positively skewed for  $\xi > 1$  and negatively skewed for  $\xi < 1$ . If  $\xi = 1$ , then for  $\beta = -1(0)[1]$  this density reduces to a uniform (Gaussian) [double-exponential] distribution.

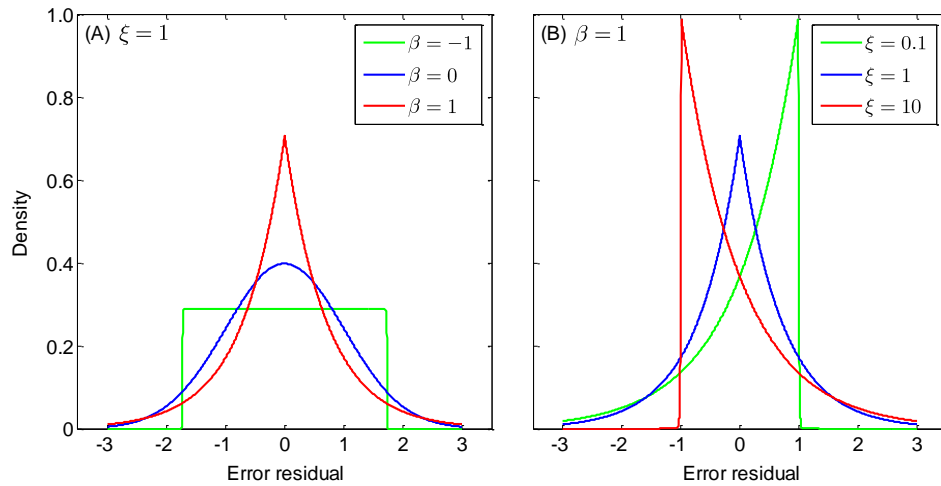


Figure A.1 Densities of the generalized likelihood function of *Schoups and Vrugt* (2010a) for different values of the kurtosis ( $\beta$ ) and skewness ( $\xi$ ).

The Student likelihood function, 17, of *Scharnagl et al.* (2015) is designed in part to better mimic residual distributions with heavy tails (see Figure ??).

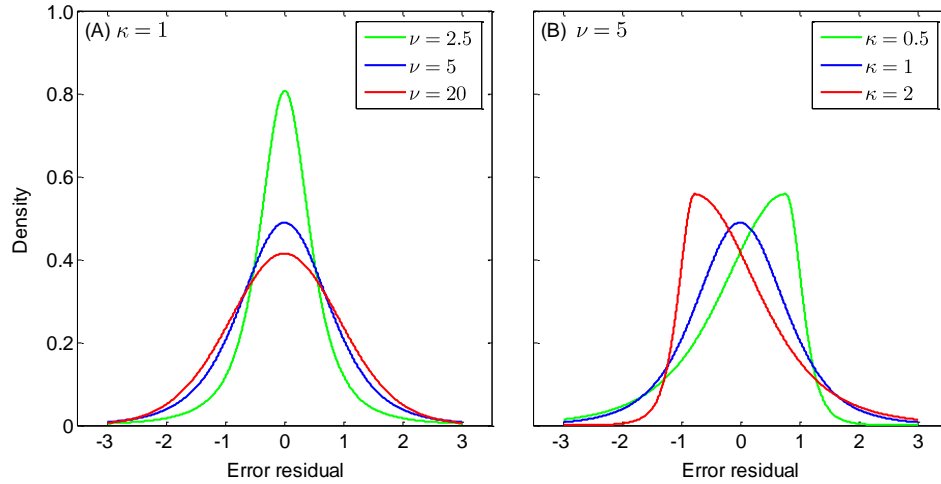


Figure A.2 Densities of the skewed Student likelihood function of *Scharnagl et al. (2015)* for different values of the skewness ( $\kappa$ ) and kurtosis ( $\xi$ ).

Table A.2 summarizes several commonly used formal likelihood functions in hydrologic modeling applications and lists how likelihood function 14 can be reduced to these by making specific assumptions about the error residuals (see also *Schoups and Vrugt (2010a)*).

Table A.2 Relationship of likelihood functions used/proposed in the hydrologic literature and the likelihood function 14 of DREAM Suite.

| Reference  | Implementation using 14  |
|--|--|
| Standard least squares                             | $\phi_1 = 0; \phi_2 = 0; \phi_3 = 0; \phi_4 = 0; \sigma_1 = 0; \xi = 1; \beta = 0$ |
| <i>Sorooshian and Dracup (1980): Equation (20)</i> | $\phi_2 = 0; \phi_3 = 0; \phi_4 = 0; \sigma_1 = 0; \xi = 1; \beta = 0$             |
| <i>Sorooshian and Dracup (1980): Equation (26)</i> | $\phi_1 = 0; \phi_2 = 0; \phi_3 = 0; \phi_4 = 0; \xi = 1; \beta = 0$               |
| <i>Kuczera (1983)</i>                              | $\beta = 0$  |
| <i>Bates and Campbell (2001)</i>                   | $\beta = 0$  |
| <i>Thiemann et al. (2001)</i>                      | $\phi_1 = 0; \phi_2 = 0; \phi_3 = 0; \phi_4 = 0$                                   |

By fixing some of the values of the nuisance variables the likelihood function can be simplified to a specific family of probability distributions.

I am now left to describe how to setup the joint inference of the model and nuisance parameters. This is actually quite simple in DREAM Suite. The nuisance variables will appear automatically once a likelihood function is selected by the user.

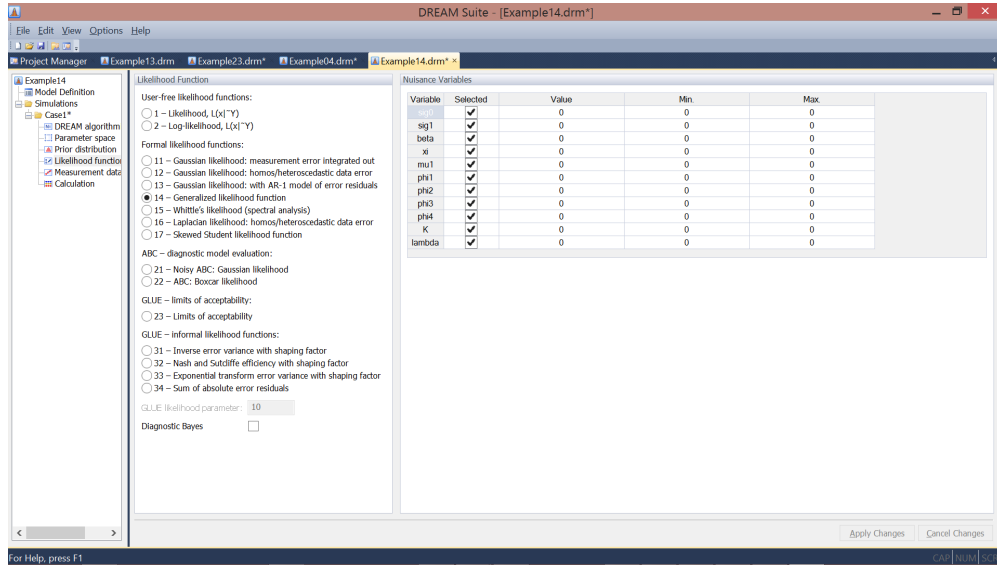


Figure A.3 Screenshot of GUI of DREAM Suite after selection of likelihood 14.

The user can select whichever nuisance variable is deemed appropriate for inference - along with its ranges. Those nuisance variables that are kept fixed will assume default values. These values are listed in the GUI but can be changed by the user.

For other likelihood functions with nuisance variables (13 and 17) a similar approach and setup is used. For instance, for likelihood 13 the user has to define the ranges of the nuisance variable  $\rho$  (or fix this variable to one value), and for likelihood 17, the nuisance variables  $\sigma_a, \sigma_b, \sigma_c, \sigma_d, \phi, \nu, \kappa$  need to be defined (or fixed) along with the values of the anchor points  $x_1, x_2, x_3$ , and  $x_4$  to be used with piecewise cubic hermite interpolation (*Scharnagl et al.*, 2015).



## Appendix B. REFERENCES

- B. Ahrens, and M. Reichstein, "Reconciling  $^{14}\text{C}$  and minirhizotron-based estimates of fine-root turnover with survival functions," *Journal of Plant Nutrition and Soil Science*, vol. 177, pp. 287-296, doi:10.1002/jpln.201300110, 2014.
- M. Barthel, A. Hammerle, P. Sturm, T. Baur, L. Gentsch, and A. Knohl, "The diel imprint of leaf metabolism on the  $\delta^{13}\text{C}$  signal of soil respiration under control and drought conditions," *New Phytologist*, vol. 192, pp. 925-938, doi: 10.1111/j.1469-8137.2011.03848.x, 2011.
- B.C. Bates, and E.P. Campbell, "A Markov chain Monte Carlo scheme for parameter estimation and inference in conceptual rainfall-runoff modeling," *Water Resources Research*, vol. 37 (4), pp. 937-947, 2001.
- L. Bauwens, B. de Backer, and A. Dufays, "Estimating and forecasting structural breaks in financial time series," Economics, Finance, Operations Research, Econometrics, and Statistics, Discussion paper, pp. 1-23, 2011.
- M.A. Beaumont, W. Zhang, and D.J. Balding, "Approximate Bayesian computation in population genetics," *Genetics*, vol. 162 (4), pp. 2025-2035, 2002.
- K.J. Beven, and A.M. Binley, "The future of distributed models: model calibration and uncertainty prediction," *Hydrological Processes*, vol. 6, pp. 279-98, 1992.
- K. Beven, and J. Freer, "Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology," *Journal of Hydrology*, vol. 249 (1-4), pp. 11-29, doi:10.1016/S0022-1694(01)00421-8, 2001.
- K. Beven, "A manifesto for the equifinality thesis," *Journal of Hydrology*, vol. 320 (1), pp. 18-36, 2006.
- K.J. Beven, and A.M. Binley, "GLUE: 20 years on," *Hydrological Processes*, vol. 28, pp. 5879-5918, 2014, doi:10.1002/hyp.10082, 2014.
- R.S. Blasone, J.A. Vrugt, H. Madsen, D. Rosbjerg, G.A. Zyvoloski, and B.A. Robinson, "Generalized likelihood uncertainty estimation (GLUE) using adaptive Markov chain Monte Carlo sampling," *Advances in Water Resources*, vol. 31, pp. 630-648, doi:10.1016/j.advwatres.2007.12.003, 2008.
- C.J.F. ter Braak, "A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces," *Statistics & Computing*, vol. 16, pp. 239-249, 2006.
- C.J.F. ter Braak, and J.A. Vrugt, "Differential evolution Markov chain with snooker updater and fewer chains," *Statistics & Computing*, vol. 18 (4), pp. 435-446, doi:10.1007/s11222-008-9104-9, 2008.
- M.C. Braakhekke, T. Wutzler, C. Beer, J. Kattge, M. Schrumpf, B. Ahrens, I. Schöning, M.R. Hoosbeek, B. Kruijt, P. Kabat, and M. Reichstein, "Modeling the vertical soil organic matter profile using Bayesian parameter estimation," *Biogeosciences*, vol. 10, pp. 399-420, doi:10.5194/bg-10-399-2013, 2013.
- J. Bikowski, J.A. Huisman, J.A. Vrugt, H. Vereecken, and J. van der Kruk, "Inversion and sensitivity analysis of ground penetrating radar data with waveguide dispersion using deterministic and Markov chain Monte Carlo methods," *Near Surface Geophysics*, vol. 10 (6), pp. 641-652, doi:10.3997/1873-0604.2012041, 2012.
- G.E.P. Box, and D.R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society, Series B*, vol. 26 (2), pp. 211-252, 1964.
- S.P. Brooks, and A. Gelman, "General methods for monitoring convergence of iterative simulations," *Journal of Computational and Graphical Statistics*, vol. 7, pp. 434-455, 1998.
- W. Chauvenet, "A Manual of spherical and practical astronomy V. II. 1863," Reprint of 1891, 5th ed. Dover, N.Y., pp. 474-566, 1960.
- S. Chib, "Marginal likelihood from the Gibbs output," *Journal of the American Statistical Association*, vol. 90, pp. 1313-1321, 1995.
- M.P. Clark, and D. Kavetski, "Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes," *Water Resources Research*, vol. 46, W10510, doi:10.1029/2009WR008894, 2010.
- F.C. Coelho, C.T. Codeço, and M.G.M. Gomes, "A Bayesian framework for parameter estimation in dynamical models," *PLoS ONE*, vol. 6 (5), e19616, doi:10.1371/journal.pone.0019616, 2011.
- K. Davis, and Y. Li, "Fast solution of geophysical inversion using adaptive mesh, space-filling curves and wavelet compression," *Geophysical Journal International*, vol. 185 (1), pp. 157-166, 2011.
- S.C. DeCaluwe, P.A. Kienzle, P. Bhargava, A.M. Baker, and J.A. Dura, "Phase segregation of sulfonate groups in Nafion interface lamellae, quantified via neutron reflectometry fitting techniques for multi-layered structures," *Soft Matter*, vol. 10, pp. 5763-5777, doi:10.1039/C4SM00850B, 2014.
- S.C. Dekker, J.A. Vrugt, and R.J. Elkington, "Significant variation in vegetation characteristics and dynamics from ecohydrologic optimality of net carbon profit," *Ecohydrology*, vol. 5, pp. 1-18, doi:10.1002/eco.177, 2010.

- P. Del Moral, A. Doucet, and A. Jasra, "An adaptive sequential Monte Carlo method for approximate Bayesian computation," *Statistics & Computing*, vol. 22, pp. 1009-1020, doi:10.1007/s11222-011-9271-y, 2012.
- A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39(B), pp. 1-39, 1977.
- D.G.T. Denison, C.C. Holmes, B.K. Mallick, and A.F.M. Smith, "Bayesian methods for nonlinear classification and regression," John Wiley & Sons, Chichester, 2002.
- P.J. Diggle, and R.J. Gratton, "Monte Carlo methods of inference for implicit statistical models," *Journal of the Royal Statistical Society Series B*, vol. 46, pp. 193-227, 1984.
- Q. Duan, J. Schaake, V. Andréassian, S. Franks, G. Goteti, H.V. Gupta, Y.M. Gusev, F. Habets, A. Hall, L. Hay, T. Hogue, M. Huang, G. Leavesley, X. Liang, O.N. Nasonova, J. Noilhan, L. Oudin, S. Sorooshian, T. Wagener, E.F. Wood, "Model Parameter Estimation Experiment (MOPEX): An overview of science strategy and major results from the second and third workshops," *Journal of Hydrology*, vol. 320, pp. 3-17, doi:10.1016/j.jhydrol.2005.07.031, 2006.
- B. Dumont, V. Leemans, M. Mansouri, B. Bodson, J.-P. Destain, M.-F. Destain, "Parameter identification of the STICS crop model, using an accelerated formal MCMC approach," *Environmental Modeling & Software*, vol. 52, pp. 121-135, 2014.
- J.A. Dura, S.T. Kelly, P.A. Kienzle, J.-H. Her, T.J. Udovic, C.F. Majkrzak, C.-J. Chung, and B.M. Clemens, "Porous Mg formation upon dehydrogenation of MgH<sub>2</sub> thin films," *Journal of Applied Physics*, vol. 109, 093501, 2011.
- G. Evin, D. Kavetski, M. Thyer, and G. Kuczera, "Pitfalls and improvements in the joint inference of heteroscedasticity and autocorrelation in hydrological model calibration," *Water Resources Research*, vol. 49, pp. 4518-4524, doi:10.1002/wrcr.20284, 2013.
- J. Freer, K.J. Beven, B. Ambroise, "Bayesian estimation of uncertainty in runoff prediction and the value of data: an application of the GLUE approach," *Water Resources Research*, vol. 32, pp. 2161-2173, 1996.
- A.E. Gelfand, and A.F. Smith, "Sampling based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, pp. 398-409, 1990.
- A.G. Gelman, and D.B. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical Sciences*, vol. 7, pp. 457-472, 1992.
- A.G. Gelman, G.O. Roberts, W.R. Gilks, "Bayesian Statistics," Oxford University Press, pp. 599-608, 1996.
- A. Gelman, and X.L. Meng, "Simulating normalizing constants: from importance sampling to bridge sampling to path sampling," *Statistical Science*, vol. 13 (2), pp. 163-185, 1998.
- A.G. Gelman, and K. Shirley, "Inference from simulations and monitoring convergence," pp. 163-174 in *The handbook of Markov chain Monte Carlo* (Eds: X.L. Meng, A. Gelman, and G. Jones), Chapman & Hall/CRC Press, 2009.
- L. Gentsch, A. Hammerle, P. Sturm, J. Ogée, L. Wingate, R. Siegwolf, P. Plüss, T. Baur, N. Buchmann, and A. Knohl, "Carbon isotope discrimination during branch photosynthesis of *Fagus sylvatica*: a Bayesian modeling approach," *Plant, Cell & Environment*, vol. 37, pp. 1516-1535, doi: 10.1111/pce.12262, 2014.
- M.Th. van Genuchten, "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils," *Soil Science Society of America Journal*, vol. 44 (5), pp. 892-898, doi:10.2136/sssaj1980.03615995004400050002x, 1980.
- J. Geweke, "Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments," in *Bayesian Statistics 4*, edited by J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, pp. 169-193, Oxford University Press, 1992.
- W.R. Gilks, G.O. Roberts, and E.I. George, "Adaptive direction sampling," *Statistician*, vol. 43, pp. 179-189, 1994.
- W.R. Gilks, and G.O. Roberts, "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 & Hall, London, U.K., 1996.
- P.J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, pp. 711-732, 1995.
- A. Grelaud, C. Robert, J. Marin, F. Rodolphe, and J. Taly, "ABC likelihood-free methods for model choice in Gibbs random fields," *Bayesian Analysis*, vol. 4 (2), pp. 317-336, 2009.
- F.E. Grubbs, "Sample criteria for testing outlying observations," *The Annals of Mathematical Statistics*, vol. 21 (1), pp. 27-58, doi:10.1214/aoms/1177729885, 1950.
- H.V. Gupta, T. Wagener, and Y. Liu, "Reconciling theory with observations: elements of a diagnostic approach to model evaluation," *Hydrological Processes*, vol. 22 (18), pp. 3802-3813, 2008.
- H. Haario, E. Saksman, J. Tamminen, "Adaptive proposal distribution for random walk Metropolis algorithm," *Computational Statistics*, vol. 14, pp. 375-395, 1999.
- H. Haario, E. Saksman, J. Tamminen, "An adaptive Metropolis algorithm," *Bernoulli*, vol. 7, pp. 223-242, 2001.

- H. Haario, E. Saksman, and J. Tamminen, "Componentwise adaptation for high dimensional MCMC," *Statistics & Computing*, vol. 20, pp. 265-274, 2005.
- H. Haario, M. Laine, A. Mira, and E. Saksman, "DRAM: Efficient adaptive MCMC," *Statistics & Computing*, vol. 16, pp. 339-354, 2006.
- T.M. Hansen, K.S. Cordua, and K. Mosegaard, "Inverse problems with non-trivial priors: efficient solution through sequential Gibbs sampling," *Computers and Geosciences*, vol. 16, pp. 593-611, 2012.
- K.W. Harrison, S.V. Kumar, C.D. Peters-Lidard, and J.A. Santanello, "Quantifying the change in soil moisture modeling uncertainty from remote sensing observations using Bayesian inference techniques," *Water Resources Research*, vol. 48, W11514, doi:10.1029/2012WR012337, 2012.
- H. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97-109, 1970.
- A.W. Hinnell, T.P.A. Ferré, J.A. Vrugt, S. Moysey, J.A. Huisman, and M.B. Kowalsky, "Improved extraction of hydrologic information from geophysical data through coupled hydrogeophysical inversion," *Water Resources Research*, vol. 46, W00D40, doi:10.1029/2008WR007060, 2010.
- V.R. Horowitz, B.J. Aleman, D.J. Christle, A.N. Cleland, and D.D. Awschalom, "Electron spin resonance of nitrogen-vacancy centers in optically trapped nanodiamonds," *Proceedings of the National Academy of the United States of America*, vol. 109 (34), pp. 13493-13497, doi:10.1073/pnas.1211311109, 2012.
- T. Iizumi, Y. Tanaka, G. Sakurai, Y. Ishigooka, and M. Yokozawa, "Dependency of parameter values of a crop model on the spatial scale of simulation," *Journal of Advances in Modeling Earth Systems*, vol. 06, doi:10.1002/2014MS000311, 2014.
- B. Jafarpour, V.K. Goyal, D.B. McLaughlin, and W.T. Freeman, "Transform-domain sparsity regularization for inverse problems in geosciences," *Geophysics*, vol. 74 (5), pp. R69-R83, 2009.
- B. Jafarpour, V.K. Goyal, D.B. McLaughlin, and W.T. Freeman, "Compressed history matching: exploiting transform-domain sparsity for regularization of nonlinear dynamic data integration problems," *Mathematical Geosciences*, 42 (1), pp. 1-27, 2010.
- B. Jafarpour, "Wavelet reconstruction of geologic facies from nonlinear dynamic flow measurements," *IEEE Transactions on Geosciences and Remote Sensing*, vol. 49 (5), pp. 1520-1535, 2011.
- J.F. Joseph, and J.H.A. Guillaume, "Using a parallelized MCMC algorithm in R to identify appropriate likelihood functions for SWAT," *Environmental Modeling & Software*, vol. 46, pp. 292-298, 2013.
- P. Joyce, and P. Marjoram, "Approximately sufficient statistics and Bayesian computation," *Statistical Applications in Genetics and Molecular Biology*, vol. 7 (1), 2008.
- R.E. Kass, and A.E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, pp. 773-795, 1995.
- D. Kavetski, G. Kuczera, and S.W. Franks, "Bayesian analysis of input uncertainty in hydrological modeling: 1. Theory," *Water Resources Research*, vol. 42 (3), W03407, doi:10.1029/2005WR004368, 2006a.
- D. Kavetski, G. Kuczera, and S.W. Franks, "Bayesian analysis of input uncertainty in hydrological modeling: 2. Application," *Water Resources Research*, vol. 42 (3), W03408, doi:10.1029/2005WR004376, 2006b.
- E.H. Keating, J. Doherty, J.A. Vrugt, and Q. Kang, "Optimization and uncertainty assessment of strongly nonlinear groundwater models with high parameter dimensionality," *Water Resources Research*, vol. 46, W10517, doi:10.1029/2009WR008584, 2010.
- W.Y. Kow, W.L. Khong, Y.K. Chin, I. Saad, K.T.K. Teo, "Enhancement of Markov chain monte Carlo convergence speed in vehicle tracking using genetic operator," 2012 Fourth International Conference on Computational Intelligence, Modeling and Simulation (CIMSIm), pp. 270-275, doi:10.1109/CIMSIm.2012.61, 2012.
- L. Krayner, J.W. Lau, and B.J. Kirby, "Structural and magnetic etch damage in CoFeB," *Journal of Applied Physics*, vol. 115, 17B751, 2014.
- G. Kuczera, "Improved parameter inference in catchment models, 1. Evaluating parameter uncertainty," *Water Resources Research*, vol. 19 (5), pp. 1151-1162, doi:10.1029/WR019i005p01151, 1983.
- G. Kuczera, D. Kavetski, S. Franks, and M. Thyer, "Towards a Bayesian total error analysis of conceptual rainfall-runoff models: Characterising model error using storm-dependent parameters," *Journal of Hydrology*, vol. 331 (1), pp. 161-177, 2006.
- G. Kuczera, D. Kavetski, B. Renard, and M. Thyer, "A limited memory acceleration strategy for MCMC sampling in hierarchical Bayesian calibration of hydrological models," *Water Resources Research*, vol. 46, W07602, doi:10.1029/2009WR008985, 2010.
- C.P. Kikuchi, T.P.A. Ferré, and J.A. Vrugt, "Discrimination-inference for measurement selection," *Water Resources Research*, vol. XX, XX-XX, doi:10.1002/wrcr.XXXX, 2015.
- B.J. Kirby, M.T. Rahman, R.K. Dumas, J.E. Davies, C.H. Lai, and K. Liu, "Depth-resolved magnetization reversal in nanoporous perpendicular anisotropy multilayers," *Journal of Applied Physics*, vol. 113, 033909, doi:10.1063/1.4775819, 2013.

- 2013.
- E. Laloy, and J.A. Vrugt, "High-dimensional posterior exploration of hydrologic models using multiple-try DREAM<sub>(ZS)</sub> and high-performance computing," *Water Resources Research*, vol. 48, W01526, doi:10.1029/2011WR010608, 2012a.
- E. Laloy, N. Linde, and J.A. Vrugt, "Mass conservative three-dimensional water tracer distribution from Markov chain Monte Carlo inversion of time-lapse ground-penetrating radar data," *Water Resources Research*, vol. 48, W07510, doi:10.1029/2011WR011238, 2012b.
- E. Laloy, B. Rogiers, J.A. Vrugt, D. Jacques, and D. Mallants, "Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov chain Monte Carlo simulation and polynomial chaos expansion," *Water Resources Research*, vol. 49 (5), pp. 2664-2682, doi:10.1002/wrcr.20226, 2013.
- E. Laloy, N. Linde, D. Jacques, and J.A. Vrugt, "Probabilistic inference of multi-Gaussian fields from indirect hydrological data using circulant embedding and dimensionality reduction," *Water Resources Research*, vol. XX, WXXXXX, doi:10.1029/2014WR0XXXXX, 2015, In Press.
- P. S. Laplace, "Mémoire sur la probabilité des causes par les évènements," *Mémoires de l'Académie Royale des Sciences Présentés par Divers Savan*, vol. 6, pp. 621-656, 1774.
- G.E. Leventhal, H.F. Günthard, S. Bonhoeffer, and T. Stadler, "Using an epidemiological model for phylogenetic inference reveals density dependence in HIV transmission," *Molecular Biology and Evolution*, vol. 31 (1), pp. 6-17, doi:10.1093/molbev/mst172, 2013.
- S.M. Lewis, and A.E. Raftery, "Estimating Bayes factors via posterior simulation with the Laplace-Metropolis estimator," *Journal of the American Statistical Association*, vol. 92 (438), pp. 648-655, 1997.
- N. Linde, and J.A. Vrugt, "Distributed soil moisture from crosshole ground-penetrating radar travel times using stochastic inversion," *Vadose Zone Journal*, vol. 12 (1), doi:10.2136/vzj2012.0101, 2013.
- J. Lise, C. Meghir, and J-M. Robin, "Mismatch, sorting and wage dynamics," National Bureau of Economic Research, Working paper, 18719, pp. 1-43, <http://www.nber.org/papers/w18719>, 2012.
- J. Lise, "On the job search and precautionary savings," *Review of economic studies*, vol. 80 (3), pp. 1086-1113, doi:10.1093/restud/rds042, 2013.
- J.S. Liu, F. Liang, and W.H. Wong, "The multiple-try method and local optimization in metropolis sampling," *Journal of the American Statistical Association*, vol. 95 (449), pp. 121-134, doi:10.2307/2669532, 2000.
- T. Lochbühler, S.J. Breen, R.L. Detwiler, J.A. Vrugt, and N. Linde, "Probabilistic electrical resistivity tomography for a CO<sub>2</sub> sequestration analog," *Journal of Applied Geophysics*, vol. 107, pp. 80-92, doi:10.1016/j.jappgeo.2014.05.013, 2014.
- T. Lochbühler, J.A. Vrugt, M. Sadegh, and N. Linde, "Summary statistics from training images as prior information in probabilistic inversion," *Geophysical Journal International*, vol. 201, pp. 157-171, doi:10.1093/gji/ggv008.
- D. Lu, M. Ye, M.C. Hill, E.P. Poeter, and G.P. Curtis, "A computer program for uncertainty analysis integrating regression and Bayesian methods," *Environmental Modeling & Software*, vol. 60, pp. 45-56, 2014.
- B. Malama, K.L. Kuhlman, and S.C. James, "Core-scale solute transport model selection using Monte Carlo analysis," *Water Resources Research*, vol. 49, pp. 3133-3147, doi:10.1002/wrcr.20273, 2013.
- L. Mari, E. Bertuzzo, L. Righetto, R. Casagrandi, M. Gatto, I. Rodriguez-Iturbe, and A. Rinaldo, "Modeling cholera epidemics: the role of waterways, human mobility and sanitation," *Journal of the Royal Society Interface*, vol. 9 (67), pp. 376-388, 2011.
- M.D. McKay, R.J. Beckman, W.J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21 (2), pp. 239-245, doi:10.2307/1268522, 1979.
- X.L. Meng, and W.H. Wong, "Simulating ratios of normalizing constants via a simple identity: A theoretical exploration," *Statistica Sinica*, vol. 6, pp. 831-860.
- N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087-1092, 1953.
- B. Minasny, J.A. Vrugt, and A.B. McBratney, "Confronting uncertainty in model-based geostatistics using Markov chain Monte Carlo simulation," *Geoderma*, vol. 163, pp. 150-622, doi:10.1016/j.geoderma.2011.03.011, 2011.
- A. Montanari, and E. Toth, "Calibration of hydrological models in the spectral domain: An opportunity for scarcely gauged basins?," *Water Resources Research*, vol. 43, W05434, doi:10.1029/2006WR005184, 2007.
- K. Mosegaard, and A. Tarantola, "Monte Carlo sampling of solutions to inverse problems," *Journal of Geophysical Research*, vol. 100 (B7), pp. 12431-12447, 1995.
- D.J. Nott, L. Marshall, and J. Brown, "Generalized likelihood uncertainty estimation (GLUE) and approximate Bayesian computation: What's the connection?," *Water Resources Research*, vol. 48 (12), doi:10.1029/2011WR011128, 2012.
- E. Oware, S. Moysey, and T. Khan, "Physically based regularization of hydrogeophysical inverse problems for improved imaging

- of process-driven systems," *Water Resources Research*, vol. 49 (10), pp. 6238-6247, 2013.
- J.E. Owejan, J.P. Owejan, S.C. DeCaluwe, and J.A. Dura, "Solid electrolyte interphase in Li-ion batteries: Evolving structures measured in situ by neutron reflectometry", *Chemistry of Materials*, vol. 24, pp. 2133-2140, 2012.
- A.B. Owen, and S.D. Tribble, "A quasi-Monte Carlo Metropolis algorithm," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 8844-8849, 2005.
- D.G. Partridge, J.A. Vrugt, P. Tunved, A.M.L. Ekman, D. Gorea, and A. Sorooshian, "Inverse modeling of cloud-aerosol interactions - Part I: Detailed response surface analysis," *Atmospheric Chemistry and Physics*, vol. 11, pp. 4749-4806, doi:10.5194/acpd-11-4749-2011, 2011.
- D.G. Partridge, J.A. Vrugt, P. Tunved, A.M.L. Ekman, H. Struthers, and A. Sorooshian, "Inverse modeling of cloud-aerosol interactions - Part II: Sensitivity tests on liquid phase clouds using Markov chain Monte Carlo simulation approach," *Atmospheric Chemistry and Physics*, vol. 12, pp. 2823-2847, doi:10.5194/acp-12-2823-2012, 2012.
- B. Peirce, "Criterion for the Rejection of Doubtful Observations," *Astronomical Journal II*, vol. 45, 1852.
- K.V. Price, R.M. Storn, and J.A. Lampinen, *Differential evolution, A practical approach to global optimization*, Springer, Berlin, 2005.
- V.C. Radu, J. Rosenthal, and C. Yang, "Learn from the thy neighbor: parallel-chain and regional adaptive MCMC," *Journal of the American Statistical Association*, vol. 104 (488), pp. 1454-1466, 2009.
- A.E. Raftery, and S.M. Lewis, "One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo," *Statistical Science*, vol. 7, pp. 493-497, 1992.
- A.E. Raftery, D. Madigan, and J.A. Hoeting, "Bayesian model averaging for linear regression models," *Journal of the American Statistical Association*, vol. 92, pp. 179-191, 1997.
- A.E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using Bayesian model averaging to calibrate forecast ensembles," *Monthly Weather Review*, vol. 133, pp. 1155-1174, 2005.
- O. Ratmann, C. Andrieu, C. Wiuf, and S. Richardson, "Model criticism based on likelihood-free inference, with an application to protein network evolution," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, pp. 1-6, 2009.
- B. Renard, D. Kavetski, G. Kuczera, M. Thyer, and S.W. Franks, "Understanding predictive uncertainty in hydrologic modeling: The challenge of identifying input and structural errors," *Water Resources Research*, vol. 46, W05521, doi:10.1029/2009WR008328, 2010.
- B. Renard, D. Kavetski, E. Leblois, M. Thyer, G. Kuczera, and S.W. Franks, "Toward a reliable decomposition of predictive uncertainty in hydrological modeling: Characterizing rainfall errors using conditional simulation," *Water Resources Research*, vol. 47 (11), W11516, doi:10.1029/2011WR010643, 2011.
- A. Rinaldo, E. Bertuzzo, L. Mari, L. Righetto, M. Blokesch, M. Gatto, R. Casagrandi, M. Murray, S.M. Vesenbeckh, and I. Rodriguez-Iturbe, "Reassessment of the 2010-2011 Haiti cholera outbreak and rainfall-driven multiseason projections," *Proceedings of the National Academy of the United States of America*, vol. 109 (17), pp. 6602-6607, 2012.
- J. Rings, J.A. Vrugt, G. Schoups, J.A. Huisman, and H. Vereecken, "Bayesian model averaging using particle filtering and Gaussian mixture modeling: Theory, concepts, and simulation experiments," *Water Resources Research*, 48, W05520, doi:10.1029/2011WR011607, 2012.
- G.O. Roberts, and W.R. Gilks, "Convergence of adaptive direction sampling," *Journal of Multivariate Analysis*, vol. 49, pp. 287-298, 1994.
- G.O. Roberts, A. Gelman, and W.R. Gilks, "Weak convergence and optimal scaling of random walk Metropolis algorithms," *Annals of Applied Probability*, vol. 7, pp. 110-120, 1997.
- C.P. Roberts, and G. Casella, "Monte Carlo statistical methods," 2nd edition, Springer, New York, 2004.
- G.O. Roberts, and J.S. Rosenthal, "Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms," *Journal of Applied Probability*, vol. 44, pp. 458-475, 2007.
- M. Rosas-Carbajal, N. Linde, T. Kalscheuer, and J.A. Vrugt, "Two-dimensional probabilistic inversion of plane-wave electromagnetic data: Methodology, model constraints and joint inversion with electrical resistivity data," *Geophysical Journal International*, vol. 196 (3), pp. 1508-1524, doi: 10.1093/gji/ggt482, 2014.
- P. Ruggeri, J. Irving, and K. Holliger, "Systematic evaluation of sequential geostatistical resampling within MCMC for posterior sampling of near-surface geophysical inverse problems," *Geophysical Journal International*, vol. 202, pp. 961-975, doi:10.1093/gji/ggv196, 2015.
- M. Sadeh, and J.A. Vrugt (2013), "Approximate Bayesian Computation in hydrologic modeling: equifinality of formal and informal approaches," *Hydrology and Earth System Sciences - Discussions*, vol. 10, 4739-4797, doi:10.5194/hessd-10-4739-



- 2013, 2013.
- M. Sadegh, and J.A. Vrugt, "Approximate Bayesian computation using Markov chain monte Carlo simulation: DREAM<sub>(ABC)</sub>," *Water Resources Research*, vol. 50, doi:10.1002/2014WR015386, 2014.
- M. Sadegh, J.A. Vrugt, and H.V. Gupta, "The soil water characteristic as new class of closed-form parametric expressions for the flow duration curve," *Water Resources Research*, vol. XX, doi:10.1002/2014WRXXXX, 2015a.
- M. Sadegh, J.A. Vrugt, C. Xu, and E. Volpi, "The stationarity paradigm revisited: Hypothesis testing using diagnostics, summary metrics, and DREAM<sub>(ABC)</sub>," *Water Resources Research*, vol. XX, pp. XX-XX, doi:10.1002/2014WRXXXXX, 2015b, In Press.
- M.G. Schaap, F.J. Leij, and M.Th. van Genuchten, "Neural network analysis for hierarchical prediction of soil water retention and saturated hydraulic conductivity," *Soil Science Society of America Journal*, vol. 62, pp. 847-855, 1998.
- M.G. Schaap, F.J. Leij, and M.Th. van Genuchten, "Rosetta: a computer program for estimating soil hydraulic parameters with hierarchical pedotransfer functions," *Journal of Hydrology*, vol. 251, pp. 163-176, 2001.
- B. Scharnagl, J.A. Vrugt, H. Vereecken, and M. Herbst, "Information content of incubation experiments for inverse estimation of pools in the Rothamsted carbon model: a Bayesian perspective," *Biogeosciences*, vol. 7, pp. 763-776, 2010.
- B. Scharnagl, J.A. Vrugt, H. Vereecken, and M. Herbst, "Bayesian inverse modeling of soil water dynamics at the field scale: using prior information about the soil hydraulic properties," *Hydrology and Earth System Sciences*, vol. 15, pp. 3043-3059, doi:10.5194/hess-15-3043-2011, 2011.
- B. Scharnagl, S.C. Iden, W. Durner, H. Vereecken, and M. Herbst, "Inverse modelling of in situ soil water dynamics: accounting for heteroscedastic, autocorrelated, and non-Gaussian distributed residuals," *Hydrology and Earth System Sciences Discussions*, vol. 12, pp. 2155-2199, 2015.
- G. Schoups, and J.A. Vrugt, "A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic and non-gaussian errors," *Water Resources Research*, vol. 46, W10531, doi:10.1029/2009WR008933, 2010a.
- G. Schoups, J.A. Vrugt, F. Fenicia, and N.C. van de Giesen, "Corruption of accuracy and efficiency of Markov chain Monte Carlo simulation by inaccurate numerical implementation of conceptual hydrologic models," *Water Resource Research*, vol. 46, W10530, doi:10.1029/2009WR008648, 2010b.
- A. Schöniger, T. Wöhling, L. Samaniego, and W. Nowak, "Model selection on solid ground: Rigorous comparison of nine ways to evaluate Bayesian model evidence," *Water Resource Research*, vol. 50 (12), W10530, pp. 9484-9513, doi:10.1002/2014WR016062, 2014.
- M. Shafii, B. Tolson, and L.S. Matott, "Uncertainty-based multi-criteria calibration of rainfall-runoff models: a comparative study," *Stochastic Environmental Research and Risk Assessment*, vol. 28 (6), pp. 1493-1510, 2014.
- J. Šimůnek, M. Šejna, and M. Th. van Genuchten, "The HYDRUS-1D software package for simulating the one-dimensional movement of water, heat, and multiple solutes in variably-saturated Media", V1.0, IGWMC-TPS-70, International Ground Water Modeling Center, Colorado School of Mines, Golden, CO, 186 pp, 1998.
- S.A. Sisson, Y. Fan, and M.M. Tanaka, "Sequential Monte Carlo without likelihoods," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104 (6), pp. 1760-1765, 2007.
- T. Smith, A. Sharma, L. Marshall, R. Mehrotra, and S. Sisson, "Development of a formal likelihood function for improved Bayesian inference of ephemeral catchments," *Water Resources Research*, vol. 46, W12551, doi:10.1029/2010WR009514, 2010.
- S. Sorooshian, and J.A. Dracup, "Stochastic parameter estimation procedures for hydrologic rainfall-runoff models: Correlated and heteroscedastic error cases," *Water Resources Research*, vol. 16 (2), pp. 430-442, 1980.
- J. Starrfelt, and Ø. Kaste, "Bayesian uncertainty assessment of a semi-distributed integrated catchment model of phosphorus transport," *Environmental Science: Processes & Impacts*, vol. 16, pp. 1578-1587, doi:10.1039/C3EM00619K, 2014.
- R. Storn, and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.
- X-L. Sun, S-C. Wu, H-L. Wang, Yu-G. Zhao, G-L. Zhang, Y.B. Man, M.H. Wong, "Dealing with spatial outliers and mapping uncertainty for evaluating the effects of urbanization on soil: A case study of soil pH and particle fractions in Hong Kong," *Geoderma*, vol. 195-196, pp. 220-233, 2013.
- B.J. Tarasevich, U. Perez-Salas, D.L. Masic, J. Philo, P. Kienzle, S. Krueger, C.F. Majkrzak, J.L. Gray, and W.J. Shaw, "Neutron reflectometry studies of the adsorbed structure of the Amelogenin, LRAP", *The Journal of Physical Chemistry B*, vol. 117 (11), pp. 3098-3109, doi:10.1021/jp311936j, 2013.
- M. Thiemann, M. Trosset, H. Gupta, and S. Sorooshian, "Bayesian recursive parameter estimation for hydrologic models,"

- Water Resources Research*, vol. 37 (10), pp. 2521-2535, doi:10.1029/2000WR900405, 2001.
- D.M. Toyli, D.J. Christle, A. Alkauskas, B.B. Buckley, C.G. van de Walle, and D.D. Awschalom, "Measurement and control of single nitrogen-vacancy center spins above 600 K," *Physical Review X*, vol. 2, 031001, doi:10.1103/PhysRevX.2.031001, 2012.
- B.M. Turner, and P.B. Sederberg, "Approximate Bayesian computation with differential evolution," *Journal of Mathematical Psychology*, vol. 56 (5), pp. 375-385, doi:10.1016/j.jmp.2012.06.004, 2012.
- G. Upton, and I. Cook, "Understanding Statistics," Oxford University Press, pp. 55, 1996.
- E. Volpi, J.A. Vrugt, and G. Schoups, "Bayesian model selection with DREAM: Multi-dimensional integration of the evidence," *Water Resources Research*, vol. XX, doi:10.1002/2014WRXXXX, 2015.
- J.A. Vrugt, H.V. Gupta, W. Bouten, and S. Sorooshian, "A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters," *Water Resources Research*, vol. 39 (8), 1201, doi:10.1029/2002WR001642, 2003.
- J.A. Vrugt, C.G.H. Diks, W. Bouten, H.V. Gupta, and J.M. Verstraten, "Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation," *Water Resources Research*, vol. 41 (1), W01017, doi:10.1029/2004WR003059, 2005.
- J.A. Vrugt, and B.A. Robinson, "Treatment of uncertainty using ensemble methods: Comparison of sequential data assimilation and Bayesian model averaging," *Water Resources Research*, vol. 43, W01411, doi:10.1029/2005WR004838, 2007a.
- J.A. Vrugt, and B.A. Robinson, "Improved evolutionary optimization from genetically adaptive multimethod search," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, pp. 708-711, doi:10.1073/pnas.0610471104, 2007b.
- J.A. Vrugt, C.J.F. ter Braak, M.P. Clark, J.M. Hyman, and B.A. Robinson, "Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation," *Water Resources Research*, vol. 44, W00B09, doi:10.1029/2007WR006720, 2008a.
- J.A. Vrugt, P.H. Stauffer, T. Wöhling, B.A. Robinson, and V.V. Vesselinov, "Inverse modeling of subsurface flow and transport properties: A review with new developments," *Vadose Zone Journal*, vol. 7 (2), pp. 843-864, doi:10.2136/vzj2007.0078, 2008b.
- J.A. Vrugt, C.G.H. Diks, and M.P. Clark, "Ensemble Bayesian model averaging using Markov chain Monte Carlo sampling," *Environmental Fluid Mechanics*, vol. 8 (5-6), pp. 579-595, doi:10.1007/s10652-008-9106-3, 2008c.
- J.A. Vrugt, C.J.F. ter Braak, C.G.H. Diks, D. Higdon, B.A. Robinson, and J.M. Hyman, "Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling," *International Journal of Non-linear Sciences and Numerical Simulation*, vol. 10, no. 3, pp. 273-290, 2009a.
- J.A. Vrugt, C.J.F. ter Braak, H.V. Gupta, and B.A. Robinson, "Equifinality of formal (DREAM) and informal (GLUE) Bayesian approaches in hydrologic modeling," *Stochastic Environmental Research and Risk Assessment*, vol. 23 (7), pp. 1011-1026, doi:10.1007/s00477-008-0274-y, 2009b.
- J.A. Vrugt, and C.J.F. ter Braak, "DREAM<sub>(D)</sub>: an adaptive Markov chain Monte Carlo simulation algorithm to solve discrete, noncontinuous, and combinatorial posterior parameter estimation problems," *Hydrology and Earth System Sciences*, vol. 15, pp. 3701-3713, doi:10.5194/hess-15-3701-2011, 2011.
- J.A. Vrugt, and M. Sadegh, "Toward diagnostic model calibration and evaluation: Approximate Bayesian computation," *Water Resources Research*, vol. 49, doi:10.1002/wrcr.20354, 2013a.
- J.A. Vrugt, C.J.F. ter Braak, C.G.H. Diks, and G. Schoups, "Advancing hydrologic data assimilation using particle Markov chain Monte Carlo simulation: theory, concepts and applications," *Advances in Water Resources*, Anniversary Issue - 35 Years, 51, 457-478, doi:10.1016/j.advwatres.2012.04.002, 2013b.
- J.A. Vrugt, "To be coherently incoherent: GLUE done with DREAM but much more accurate and efficient," *Journal of Hydrology*, vol. XX, doi:XX/XX.XX, 2015a.
- J.A. Vrugt, "The scientific method, Bayes theorem, diagnostic model evaluation, and summary metrics as prior information," *Water Resources Research*, vol. XX, doi:10.1002/wrcr.XXXXX, 2015b.
- J.A. Vrugt, "Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB Implementation," *Environmental Modeling & Software*, vol. 75, pp. 273-316, 10.1016/j.envsoft.2015.08.013, 2016.
- P. Whittle, "Estimation and information in stationary time series," *Arkiv för Matematik*, vol. 2, pp. 423-434, 1953.
- C.K. Wikle, and M.B. Hooten, "A general science-based framework for dynamic spatio-temporal models," *Test*, vol. 19, pp. 417-451, doi:10.1007/s11749-010-0209-z, 2010.
- T. Wöhling, and J.A. Vrugt, "Multi-response multi-layer vadose zone model calibration using Markov chain Monte Carlo

- simulation and field water retention data," *Water Resources Research*, vol. 47, W04510, doi:10.1029/2010WR009265, 2011.
- C.G. Yale, B.B. Buckley, D.J. Christle, G. Burkard, F.J. Heremans, L.C. Bassett, and D.D. Awschalom, "All-optical control of a solid-state spin using coherent dark states," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110 (19), pp. 7595-7600, doi:10.1073/pnas.1305920110, 2013.
- S. Zaoli, A. Giometto, M. Formentin, S. Azaele, A. Rinaldo, and A. Maritan, "Phenomenological modeling of the motility of self-propelled microorganisms," *arXiv*, 1407.1762, 2014.
- L. Zhang, and V.P. Singh, "Bivariate rainfall and runoff analysis using entropy and copula theories," *Entropy*, vol. 14, pp. 1784-1812, doi:10.3390/e14091784, 2012.
- C. Zilliox, and Frédéric Gosselin, "Tree species diversity and abundance as indicators of understory diversity in French mountain forests: Variations of the relationship in geographical and ecological space," *Forest Ecology and Management*, vol. 321 (1), pp. 105-116, 2014.