DREAM ERRATUM

Markov chain Monte Carlo Simulation Using the DREAM Software Package: Erratum

Jasper A. Vrugt[a,b,c]

[a] *Department of Civil and Environmental Engineering, University of California Irvine, 4130 Engineering Gateway, Irvine, CA 92697-2175*
[b] *Department of Earth System Science, University of California Irvine, Irvine, CA*
[c] *Institute for Biodiversity and Ecosystem Dynamics, University of Amsterdam, Amsterdam, The Netherlands*

Abstract

In a recent paper published in Environmental Modeling & Software (*Vrugt*, 2016), I introduced a MATLAB toolbox of the DiffeRential Evolution Adaptive Metropolis (DREAM) algorithm developed by *Vrugt et al.* (2008a, 2009a). This algorithm is used widely to solve Bayesian inference problems in fields ranging from physics, chemistry and engineering, to ecology, hydrology, and geophysics. This manuscript serves as an epilogue to the original DREAM toolbox paper published by *Vrugt* (2016) and documents the changes that have been made to the original MATLAB code. Most of these modifications are in response to users that request new options, whereas others changes involve removal of certain built-in variables, as their use has become obsolete.

Keywords: Bayesian inference, Markov chain Monte Carlo (MCMC) simulation, Differential Evolution Adaptive Metropolis (DREAM) algorithm

Email address: jasper@uci.edu (Jasper A. Vrugt)
URL: http://faculty.sites.uci.edu/jasper (Jasper A. Vrugt)

DREAM ERRATUM

1. ERRATUM

All the changes that are documented herein are in reference to the source code of DREAM described in *Vrugt* (2016). I use two different subsections to detail the changes that have been made. The first sub section introduces new options that have been added to the source code. The second subsection details the options that have become obsolete and therefore have been removed. Note, the changes discussed herein bave been made to the source codes of all the different algorithms of the DREAM family disuccsed in Section 7 of *Vrugt* (2016).

1.1. New options

Upon request of a user of the DREAM$_{(ZS)}$ algorithm, a new option has been added in December 2015 to the field `initial` of `Par_info`. This field specifies with a string enclosed between quotes how to sample the initial state of each of the N different Markov chains. The existing four options (1) 'uniform' (2) 'latin' (3) 'normal' and (4) 'prior' are now augmented with (5) 'user'. This new option allows the user to explicate the initial state of each Markov chain. If this new option is exercised then the user has to define in a $N \times d$ matrix the initial state (per row) of each of the N Markov chains. This matrix is called `x0` and should be stored as field of structure `Par_info`, for instance

```
Par_info.x0 = rand(N,d)
```
(1)

where N denotes the number of Markov chains that is used by DREAM to solve for the target distribution, and d equals the number of parameters of this distribution. The values of these two variables are stored in field `N` and `d` of structure `DREAMPar`, respectively. The field `x0` is thus a new field of `Par_info` and necessary if `Par_info.initial` = 'user'.

The postprocessor (function POSTPROC_DREAM) has been integrated into the main function DREAM. This has a few practical advantages. The user can deactivate screen writing of tables and figures by setting the field

print of structure options to 'no'. This field has been added to the recent version of DREAM and allows the user to control the postprocessor. The default setting of print is 'yes'. This setting will be assumed if the user does not specify the field print (or its content) of input argument options.

The latest release of DREAM also uses the latex text interpreter in all figures printed to the screen. I believe this looks better, but this might be subjective. What is more, the postprocessor (figures and tables) now use the labels $x_1, \ldots, x_d$ to indicate each of the dimensions of the posterior distribution.

A new input argument has been added to the main function of the DREAM toolbox, namely DREAM which allows the user to port other information to the plugin model/function. Thus the new function call to DREAM now becomes

```
[chain,output,fx] = DREAM(Func_name,DREAMPar,Par_info,
                                    Meas_info,options,plugin)   (2)
```

where Func_name (string), DREAMPar (structure array), and Par_info (structure array) are input arguments defined by the user, and chain (matrix), output (structure array) and fx (matrix) are output variables computed by DREAM and returned to the user. The last three input arguments of the DREAM subroutine (function), Meas_info, options and plugin are optional.

The content of the different input variables has been defined in the main manual of DREAM - which has been published in *Vrugt* (2016). Interested readers are referred to this publication for further details. The new input argument plugin is optional and allows user to pass to their model (stored as Func_name) additional information as follows

```
Y = model(x,plugin),                                            (3)
```

where Y signifies the likelihood (DREAMPar.lik = 1), the log-likelihood (DREAMPar.lik = 2), model simulation ((DREAMPar.lik = {11-17,31-34}) or vector of summary statistics (DREAMPar.lik = {21-23}). The content of plugin can be determined by the user. It can be declared a scalar, vector, matrix, structure, string or cell-array, whatever is deemed appropriate for the model plugin.

The revised DREAM codes also allows the user to define the length of the burn-in period that is used to monitor convergence via the scale-reduction diagnostics of *Gelman and Rubin* (1992) and *Brooks and Gelman* (1998). The field burnin of structure options stipulates the burn-in percentage. The default value of this field is 50, that means that only the second half of the sampled chains is used to compute both scale-reduction convergence diagnostics. A value of options.burnin = 80 would use only the last 20% of the chain to calculate the $\hat{R}_j$-diagnostic of *Gelman and Rubin* (1992), where $j = \{1, \ldots, d\}$ and the $\hat{R}^d$-statistic of *Brooks and Gelman* (1998). Furthermore, a new field logprior of structure DREAMPar enables the user to work directly with a log-density prior rather than the density of DREAMPar.prior. The content of DREAMPar.logprior is identical to DREAMPar.prior except that logprior expects the output of the prior distribution to be a log-density rather than density. The use of a log-density is encouraged, to avoid numerical underflow (prior goes to zero) in large-dimensional search spaces.

I also made two recent changes to DREAM$_{\mathrm{D}}$ (discrete estimation) that were not documented in the original published DREAM manual (*Vrugt*, 2016). First, a new field called combinatorial has been added to the structure DREAMPar and allows the user to specify with 'yes' whether the inference involves combinatorial estimation or not. The default setting of this field is 'no'. The sudoku puzzle in case study 1 of *Vrugt et al.* (2011) constitutes an example of a combinatorial parameter estimation problem, and is part of the DREAM$_{\mathrm{D}}$ software toolbox (see example 24). Another field called sort has been added to DREAMPar and allows the user to solve discrete problems for which the order of the parameter values does not matter as long as the

"right" values are included in the parameter vector. Measurement selection is such an example (see case study 2 of *Vrugt et al.* (2011); included as example 25 in MATLAB toolbox of DREAM$_{\mathrm{D}}$) as the order in which the water retention measurements are selected will not matter for the results of the inference (only the measurement set counts). Note, that detailed balance cannot be proven if the parameter vector is sorted in ascending order (low to high) after a proposal is created. Nevertheless, case studies demonstrate that the DREAM$_{\mathrm{D}}$ code converges properly.

Finally, and if so desired, the users can request a script for postprocessing outside the main DREAM programs when `options.print` = 'no'. This script will take the output of the program and generate automatically the same figures as when `options.print` = 'yes'.

1.2. Obsolete options

The new source code of DREAM automatically detects which hardware system is used (PC, Mac or Linux) and automatically configures the computational environment in MATLAB. Hence, the field `linux` of structure `options` has become obsolete and has been removed.

2. REFERENCES

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.

A.G. Gelman, and D.B. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical Sciences*, vol. 7, pp. 457-472, 1992.

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, 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 Nonlinear Sciences and Numerical Simulation*, vol. 10, no. 3, pp. 273-290, 2009a.

J.A. Vrugt, and C.J.F. ter Braak, "DREAM$_{(D)}$: 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, "Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation," *Environmental Modeling & Software*, vol. 75, 273-316, doi:10.1016//j.envsoft.2015.08.013, 2016.