Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 11, n. 1, 2025.

Trabalho apresentado no XLIII CNMAC, Centro de Convenções do Armação Resort - Porto de Galinhas - PE, 2024

DESk-McMC: Differential Evolution Markov Chain Monte Carlo with Selection Mechanisms

Michel Tosin[1], Marcio R. Borges[2]
Laboratório Nacional de Computação Científica LNCC, Petrópolis, RJ

Abstract. Metropolis is still one of the most popular algorithms used in the Bayesian analysis of stochastic problems. It is often used when the *a priori* knowledge of the target distribution is quite limited. However, the shape and size of the proposal distribution are known to be very crucial for the convergence of the algorithm. For example, the classical random-walk jump can often face convergence problems. In that sense, Differential Evolution Markov chain Monte Carlo (DE) is an interesting alternative but can also have a low acceptance rate. Inspired by genetic algorithm concepts, this work presents a new version of the DE algorithm in which a selection step is introduced. The new methodology, DESk-McMC, is applied to a simple Bayesian inference problem identified here as a polynomial "Black Box". Different values of selection pressure are studied. The results showed that the inclusion of the selection step significantly increased the average acceptance rate of Markov chains.

Keywords. Markov chain Monte Carlo, Differential Evolution, Bayesian Analysis, Random Walk, Metropolis Algorithm

1 Introduction

Recent increases in high-performance computing allied to better acquisition of dynamic flow data have attracted interest in applying Bayesian methods to characterize and reduce the uncertainties inherent in stochastic modeling, which is present in several science areas. This is particularly desired for porous media fields in Geosciences, where the stochastic dimension can be massive [1]. The Bayesian inference is convenient in quantifying the added value of information from several sources. At the same time, Markov chain Monte Carlo (McMC) methods provide a computational framework for sampling from the *a posteriori* distribution.

The Metropolis algorithm [2, 3] and its variants are an important class of McMC algorithms widely used in the Bayesian analysis of stochastic inverse problems. Despite their high computational cost, which, in some cases, can be prohibitive, McMC methods are regarded as the gold standard technique for Bayesian inference [4].

It is well known that to ensure the computational efficiency of McMC algorithms, one should choose the proposal distribution so that sampling from it would be fast and easy. Then, the shape and size of the proposal distribution are known to be very crucial for the convergence of the Markov chains [5, 6]. The standard random-walk jump is simple to understand and use but performs poorly if the target distribution has unusual shape properties. Simple mixtures of Gaussian distributions are simple examples that confuse the Random-Walk algorithm, leading to wrong distributions.

Ter Braak [7] developed an improvement in the Metropolis algorithm by incorporating Differential Evolution (DE) genetic algorithm ideas using multiple chains that are simulated in parallel.

[1] michelt@posgrad.lncc.br
[2] mrborges@lncc.br

DOI: 10.5540/03.2025.011.01.0376

In their method, information is exchanged among the multiple chains, which yields an appropriate scale and orientation for the jumping distribution. Multiple chains initiated from overdispersed starting points can be used to monitor their convergence to the target distribution. The convergence assessment is a critical step in McMC analysis, ensuring the reliability of the sampled posterior distribution.

The present work introduces a novel customized scheme called Differential Evolution Markov chain Monte Carlo with Selection, where a selection mechanism is included in the original DE method. The technique is tested in a polynomial "Black-Box" model to identify how the selective pressure affects convergence in terms of the acceptance rate [8] and Gelman-Brooks $\widehat{R}$ convergence diagnoses [9].

2 Markov chain Monte Carlo method (McMC)

The Metropolis algorithm was initially introduced by [2] for computing properties of substances composed of interacting individual molecules (when a symmetric proposal is used). [3] introduced a generalization to non-symmetric proposals. This algorithm has been widely used in several areas of science.

Let $\pi(\cdot)$ be the target distribution (distribution *a posteriori*) and $\mathsf{q}(\boldsymbol{\theta}^t, \boldsymbol{\theta})$ the instrumental proposal distribution, the Metropolis algorithm is given in Algorithm 1.

Algorithm 1 Metropolis McMC Algorithm [2]

```
procedure METROPOLIS(MaxIter)                    ▷ MaxIter: maximum number of iterations
    Initialization: Generate the initial state $\boldsymbol{\theta}^1$ from a priori distribution
    for $t = 1$ to MaxIter do
        Step 1. At state $\boldsymbol{\theta}^t$ generate $\boldsymbol{\theta}$ from the proposal distribution $\mathsf{q}(\boldsymbol{\theta}^t, \boldsymbol{\theta})$
        Step 2. Take the new state as
            $\boldsymbol{\theta}^{t+1} = \begin{cases} \boldsymbol{\theta}, & \text{with probability } \alpha(\boldsymbol{\theta}^t, \boldsymbol{\theta}) \\ \boldsymbol{\theta}^t, & \text{with probability } 1 - \alpha(\boldsymbol{\theta}^t, \boldsymbol{\theta}) \end{cases},$    (1)
        where $\alpha(\boldsymbol{\theta}^t, \boldsymbol{\theta}) = \min\left\{1, \frac{\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}^t)}\right\}$.
    end for
    return $\{\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^{\mathsf{MaxIter}+1}\}$
end procedure
```

Next, we present three ways to propose the next step in the chain (proposal distribution).

2.1 Random-Walk Metropolis (RW)

From a starting point $\boldsymbol{\theta}^1$, the Metropolis algorithm allows iteratively to produce and insert new samples in a chain from the target based on acceptance criteria [2, 3]. The classic approach for proposal is the Random-Walk where the jumping distribution is centered at the current d-dimensional point $\boldsymbol{\theta}^t$ then the proposal is given by

$$\boldsymbol{\theta} = \boldsymbol{\theta}^t + c\boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\theta}^t$ is the current element of the chain (or t-th element) and $\boldsymbol{\epsilon} \sim \mathbb{N}(0, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma}$ representing the covariance matrix. Previous works by [8] and [6] described how to choose the jump step c and reveal the optimal choice as $c = 2.38/\sqrt{d}$ and $\boldsymbol{\Sigma} = \mathsf{cov}(\boldsymbol{\theta})$ (in the case of Gaussian targets and

Gaussian proposals). The correct covariance for the target distribution is generally unknown but can be estimated during the iterative process. However, approximating the covariance matrix may be costly for high dimensions, and there is no guarantee that it will be symmetric and positively defined. In most cases, in the absence of better information, the covariance of proposal distribution is taken as the $d \times d$ identity matrix ($\mathbf{\Sigma} = \mathbb{I}_d$), and c is determined experimentally.

2.2 Differential Evolution Metropolis (DE-McMC)

Consider N_c d-dimensional parameters $\boldsymbol{\theta}_j^t, j = 1, 2, \ldots, N_c$, as members of a population $\mathbf{X}^t$ at state t. The newly drawn samples for the chain j will be obtained through the expression

$$\boldsymbol{\theta}_j = \boldsymbol{\theta}_j^t + \gamma(\boldsymbol{\theta}_{r_1}^t - \boldsymbol{\theta}_{r_2}^t) + \boldsymbol{e}\,, \tag{3}$$

where $\boldsymbol{e}$ is drawn from a symmetric distribution with a small variance compared to that of the target one but with unbounded support. The simplest is $\boldsymbol{e} \sim \mathbb{N}(0, \eta\mathbb{I}_n)$ with η small. Here, $\boldsymbol{\theta}_{r_1}^t$ and $\boldsymbol{\theta}_{r_2}^t$ are randomly selected without replacement from $\mathbf{X}^t/\{\theta_j^t\}$, i.e. $j \neq r_1 \neq r_2 \neq j$. In resume, the main factor of the DE-McMC is to use the exchange of information between the chains. The critical value for γ is $2.38/\sqrt{2d}$. Again, details can be found in [8].

2.3 Differential Evolution with Selection (DESk-McMC)

Several study cases performed by the authors of the present work have shown how DE is more efficient than RW in exploring the parameter space [7, 10]. In particular when the dimension increases. Significant modifications of the DE method were proposed by [10, 11] and others. This work proposes a less intrusive modification. The novel method, called Differential Evolution with selection (or just DESk-McMC), introduces a selection step in the choice of candidates r_1 and r_2 (Eq. 3). For this, we use the tournament scheme to randomly choose $k \geqslant 2$ elements of the current iteration among the different chains. From this set of size k, we selected r_1 and r_2 as the two highest-ranked individuals based on a fitness function. Note that if $k = 2$, the original DE-McMC is recovered. Naturally, the value of k controls the selective pressure and must be carefully defined to avoid destroying the variability ($k \ll N_c$). Here, the fitness function is based on the relative error given by Eq. 8.

3 Efficiency and convergence metrics

Efficiency is a problem dependency measure, but it is always related to the frequency with which the proposed algorithm works as desired. In the Metropolis algorithm, this is reflected in the ability to create samples that are probable to be accepted. If N_A moves are accepted from a total number of N_T (per chain, after eliminating the burning state), a measure of percentage acceptance rate can be defined as follows

$$\widehat{AR} = \frac{N_A}{N_T} \times 100\,. \tag{4}$$

For the main convergence diagnostic metric it will be applied the Gelman-Brooks d-dimensional multivariate potential scale reduction factor $\widehat{R}$ (or MPSRF) [9, 12]. Then, the convergence of the chains to the stationary distribution is announced when the $\widehat{R}$ is sufficiently close to 1. What close means varies from paper to paper, but values bigger than 1.2 are commonly unacceptable [12].

4 Polynomial "Black-Box" application ($\mathcal{BB}$)

In order to test the methodologies presented in Section 2, we create a polynomial "black box" to emulate a Bayesian inference problem. The "black box" plays the role of a simulator in which a 12-dimensional parameter is used as input, and three sets of data are produced as a result. Let $\boldsymbol{\theta} = [a_1, \dots, a_{12}]^{\mathsf{T}}$ and $x = [-1 : 0.1 : 1]$ the polynomial $\mathcal{BB}$ is defined as

$$\mathcal{BB}(x, \boldsymbol{\theta}) = \begin{cases} y_1(x) &= a_1x^3 + a_2x^2 + a_3x + a_4, \\ y_2(x) &= a_5x^3 + a_6x^2 + a_7x + a_8, \\ y_3(x) &= a_9x^3 + a_{10}x^2 + a_{11}x + a_{12} \end{cases} . \tag{5}$$

Then, for each pair $(x, \boldsymbol{\theta})$ three data sets are produced (11 points (x, y_i)). The data used as reference, $\mathcal{D}^{\mathsf{ref}} = \left[y_1^{\mathsf{ref}}, y_2^{\mathsf{ref}}, y_3^{\mathsf{ref}}\right]$, was synthetically produced as follows:

$$\mathcal{D}^{\mathsf{ref}} = \mathcal{BB}(x, \boldsymbol{\theta}^{\mathsf{ref}}) + \boldsymbol{\varepsilon}, \tag{6}$$

here, $\boldsymbol{\theta}^{\mathsf{ref}} = [40, -3, 5, 12.5, 8, -25, 2.5, 35, -40, 25, -20, 60]^{\mathsf{T}}$ and $\boldsymbol{\varepsilon} \sim \mathbb{N}(\mathbf{0}, 10^{-3} \cdot \mathbb{I}_3)$. Now, let $\mathcal{D}^{\mathsf{sim}} = \mathcal{BB}(x, \boldsymbol{\theta})$ be the simulated data, define the relative error as

$$\mathsf{E}(\boldsymbol{\theta}) = \frac{||\mathcal{D}^{\mathsf{sim}} - \mathcal{D}^{\mathsf{ref}}||^2}{||\mathcal{D}^{\mathsf{ref}}||^2} = \sum_{i=1}^{3} \frac{||y_i - y_i^{\mathsf{ref}}||^2}{||y_i^{\mathsf{ref}}||^2}. \tag{7}$$

Finally, assuming that the error between the reference and simulated data follows a Gaussian distribution, the likelihood function $\pi(\boldsymbol{\theta})$ is approximated as

$$\pi(\boldsymbol{\theta}) = \exp\left(-\frac{\mathsf{E}(\boldsymbol{\theta})}{\sigma^2}\right), \tag{8}$$

where, $\sigma = 0.01$ is the precision.

4.1 Experimental results

To solve the inverse stochastic problem given in Section 4, the RW, RWcov, DE, and DESk methods were used. RWcov means that the random walk was used; however, the proposed covariance matrix was updated every $N_b = 10000$ iteration using a random sample of size 5000.

To avoid dimensionality and convergence problems, we chose to consider 40 parallel chains and 1 million iterations. Moreover, the starting points are sorted from a uniform $\mathbb{U}(-100, 100)$ to create a well-spread start. For the random-walk scheme is used the step of $c = 2.38/\sqrt{12} \approx 0.687$ and $\epsilon \sim \mathbb{N}(\mathbf{0}, \mathbb{I}_{12})$. On the other hand, the DE-like methods utilize the scaling factor $\gamma = 2.38/\sqrt{24} \approx 0.486$ and $e \sim \mathbb{N}(\mathbf{0}, \eta\mathbb{I}_{12})$, with $\eta = 10^{-4}$. The fitness function for the DESk-McMC was a simple proportionality expression that returns higher values when the error is low. Furthermore, a doubled jump step will be used on each 10th new iteration for all the simulations.

The results are summarized in the Table 1. In all the cases, the $\widehat{R}$ and $\widehat{AR}$ values are also calculated using N_b size batches. Except for RW, convergence of the chains was observed for all other cases.

To characterize the marginal distributions, the first half of the 1 million total iterations of all chains are removed as the burning period. From the 500000×40 remaining part, N_b elements are randomly sampled to represent the posterior distribution. With this sample, histograms for each component of the parameter are constructed. The posterior distributions obtained for all methods are very similar. All components of the parameter vector (for all methods) have a distribution close to Gaussian. A Kolmogorov-Smirnov test was performed with 0.05 of significance, and the normality hypothesis was not rejected in any case. For visualization, Figure 1 displays the results

DOI: 10.5540/03.2025.011.01.0376

from the DESk-McMC with 9 competitors (or just DESk-9). The histograms (Figures 1c, 1d and 1e) are consistent with the estimated normal distributions associated. In addition, the $\widehat{R}$ curve reveals how its value floated during the simulation but never was far from the 1.01 reference value.

Even in this didactic example, the DE method performed better than RW, even when adjusting the jump covariance (RWcov). The results corroborate those reported by [7]. Here, performance is first evaluated by the convergence criterion and then by the acceptance rate. Considering that all DE-type methods converged quickly, only the acceptance rate was used to measure efficiency among them. The results show that the newly proposed methodology increased the acceptance rate as the value of k increased (up to a certain point). Additional simulations identified that this phenomenon persists up to $k = 23$, with $\widehat{AR} = 38.08\%$ (Figure 2). However, high values for k can lead to exaggerated selection pressure, and some unwanted biases can be introduced. It seems reasonable that the critical value of k is at least related to the number of chains N_c. We recommend using smaller values in the regions with the highest derivative of the curve shown in Figure 2.

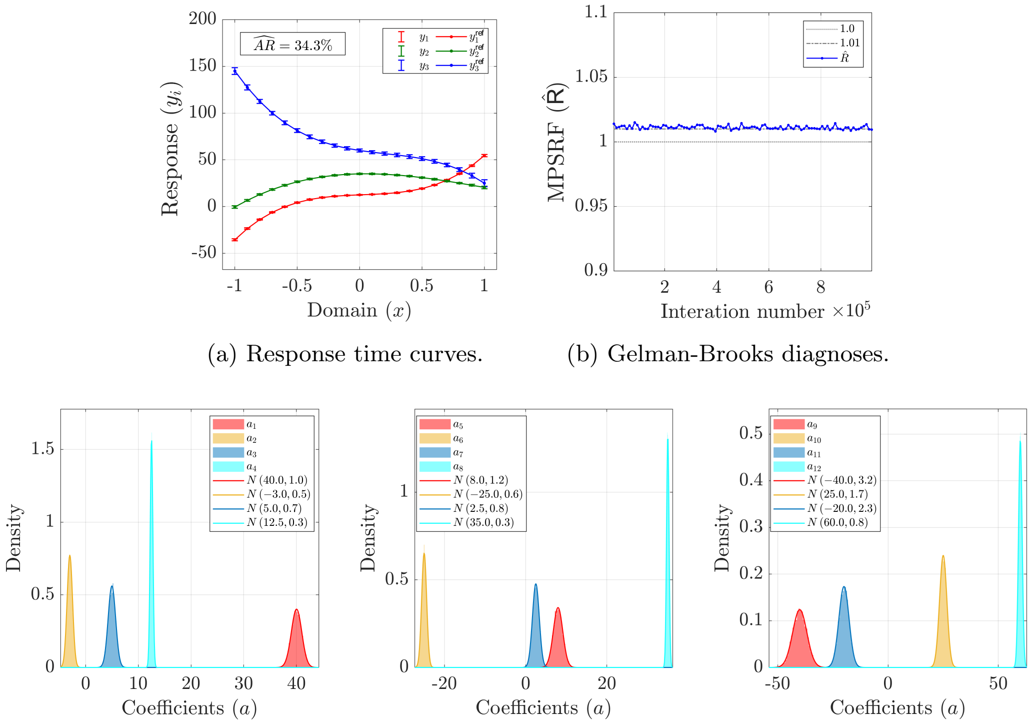
5 Final remarks

This paper introduced the novel DESk Metropolis algorithm. Results on a 12-dimensional polynomial "black box" problem showed that the new approach substantially increases the simulations' acceptance rate. Additionally, the experiments induce the existence of a critical k value which is probably related to the number of chains used in the simulation. On the other hand, high selective pressures reduce the variability at the beginning, affecting convergence. In future works, the authors intend to better investigate those characteristics and their theoretical bases. Beyond that, it is an important goal to apply the methodology to more complex problems where the high dimension and expensive function evaluations demand more efficient algorithms. In particular, the authors are interested in permeability field problems in Geosciences [1] which actually motivated this study to start.

Table 1: Simulation results for the Black-Box problem using 1000000 iterations and 40 chains.

Estimate	RW	RWcov	DE	DESk			
				3	5	7	9
$\widehat{a}_1$	$\mathbb{N}(40, 1)$	$\mathbb{N}(40, 0.9)$	$\mathbb{N}(40, 1)$	$\mathbb{N}(40, 1)$	$\mathbb{N}(40, 1)$	$\mathbb{N}(40, 1)$	$\mathbb{N}(40, 1)$
$\widehat{a}_2$	$\mathbb{N}(-3, 0.5)$	$\mathbb{N}(-3, 0.5)$	$\mathbb{N}(-3, 0.5)$	$\mathbb{N}(-3, 0.5)$	$\mathbb{N}(-3, 0.5)$	$\mathbb{N}(-3, 0.5)$	$\mathbb{N}(-3, 0.5)$
$\widehat{a}_3$	$\mathbb{N}(5.0, 0.7)$	$\mathbb{N}(5.0, 0.7)$	$\mathbb{N}(5.0, 0.7)$	$\mathbb{N}(5, 0.7)$	$\mathbb{N}(5, 0.7)$	$\mathbb{N}(5, 0.7)$	$\mathbb{N}(5, 0.7)$
$\widehat{a}_4$	$\mathbb{N}(12.5, 0.2)$	$\mathbb{N}(12.5, 0.2)$	$\mathbb{N}(12.5, 0.2)$	$\mathbb{N}(12.5, 0.3)$	$\mathbb{N}(12.5, 0.3)$	$\mathbb{N}(12.5, 0.3)$	$\mathbb{N}(12.5, 0.3)$
$\widehat{a}_5$	$\mathbb{N}(8, 1.1)$	$\mathbb{N}(8, 1.1)$	$\mathbb{N}(8, 1.1)$	$\mathbb{N}(8, 1.1)$	$\mathbb{N}(8, 1.2)$	$\mathbb{N}(7.9, 1.2)$	$\mathbb{N}(8, 1.2)$
$\widehat{a}_6$	$\mathbb{N}(-25, 0.6)$	$\mathbb{N}(-25, 0.6)$	$\mathbb{N}(-25, 0.6)$	$\mathbb{N}(-25, 0.6)$	$\mathbb{N}(-25, 0.6)$	$\mathbb{N}(-25, 0.6)$	$\mathbb{N}(-25, 0.6)$
$\widehat{a}_7$	$\mathbb{N}(2.5, 0.8)$	$\mathbb{N}(2.5, 0.8)$	$\mathbb{N}(2.5, 0.8)$	$\mathbb{N}(2.5, 0.8)$	$\mathbb{N}(2.5, 0.8)$	$\mathbb{N}(2.5, 0.8)$	$\mathbb{N}(2.5, 0.8)$
$\widehat{a}_8$	$\mathbb{N}(35, 0.3)$	$\mathbb{N}(35, 0.3)$	$\mathbb{N}(35, 0.3)$	$\mathbb{N}(35, 0.3)$	$\mathbb{N}(35, 0.3)$	$\mathbb{N}(35, 0.3)$	$\mathbb{N}(35, .0.3)$
$\widehat{a}_9$	$\mathbb{N}(-40, 3.1)$	$\mathbb{N}(-40.1, 3.1)$	$\mathbb{N}(-40, 3.1)$	$\mathbb{N}(-40, 3.2)$	$\mathbb{N}(-40, 3.2)$	$\mathbb{N}(-40, 3.2)$	$\mathbb{N}(-40, 3.2)$
$\widehat{a}_{10}$	$\mathbb{N}(25, 1.6)$	$\mathbb{N}(25, 1.6)$	$\mathbb{N}(25, 1.6)$	$\mathbb{N}(25, 1.7)$	$\mathbb{N}(25, 1.6)$	$\mathbb{N}(25.1, 1.7)$	$\mathbb{N}(25, 1.7)$
$\widehat{a}_{11}$	$\mathbb{N}(-20, 2.2)$	$\mathbb{N}(-20, 2.2)$	$\mathbb{N}(-20, 2.2)$	$\mathbb{N}(-20, 2.3)$	$\mathbb{N}(-19.9, 2.3)$	$\mathbb{N}(-20, 2.3)$	$\mathbb{N}(-20, 2.3)$
$\widehat{a}_{12}$	$\mathbb{N}(60, 0.8)$	$\mathbb{N}(60, 0.8)$	$\mathbb{N}(60, 0.8)$	$\mathbb{N}(60, 0.8)$	$\mathbb{N}(60, 0.8)$	$\mathbb{N}(60, 0.8)$	$\mathbb{N}(60, 0.8)$
$\widehat{AR}$	1.0	18.8	23.5	26.6	30.5	32.8	34.3

DOI: 10.5540/03.2025.011.01.0376



(a) Response time curves.

(b) Gelman-Brooks diagnoses.

(c) Histograms for a_j.

(d) Histograms for a_j.

(e) Histograms for a_j.

Figure 1: Result from the DESk-9 scenario with 40 chains and 1 million iterations. In (a) the deviations are doubled for better visualization and the acceptance rate was embedded to complement the general understanding of the results. Source: From authors.

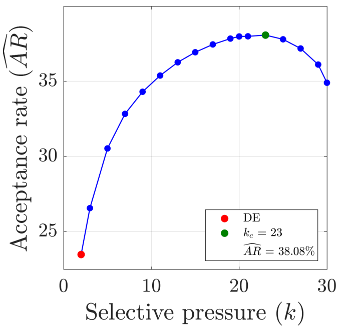


Figure 2: Acceptance rate from the DESk-McMC as a function of the parameter k. Source: From authors.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 88887.932390/2024-00.

References

[1] M. R. Borges and F. Pereira. “A novel approach for subsurface characterization of coupled fluid flow and geomechanical deformation: the case of slightly compressible flows”. In: **Computational Geosciences** 24 (2020), pp. 1693–1706.

[2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A.H. Teller, and E. Teller. “Equation of state calculations by fast computing machines”. In: **The Journal of Chemical Physics** 21 (1953), pp. 1087–1092.

[3] W. K. Hastings. “Monte carlo sampling methods using markov chains and their applications”. In: **Biometrika** 57 (1970), pp. 97–109.

[4] C. Nemeth and P. Fearnhead. “Stochastic Gradient Markov Chain Monte Carlo”. In: **Journal of the American Statistical Association** 116.533 (2021), pp. 433–450.

[5] H. Haario, E. Saksman, and J. Tamminen. “Adaptive proposal distribution for random walk Metropolis algorithm”. In: **Computational Statistics** 14.3 (1999), pp. 375–395.

[6] A. Gelman, G. O. Roberts, and W. R. Gilks. “Efficient Metropolis Jumping Rules”. In: **Bayesian Statistics**. Ed. by J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith. Oxford University Press, Oxford, 1996, pp. 599–608.

[7] C. J. F. T. Braak. “A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces”. In: **Statistics and Computing** 16 (2006), pp. 239–249.

[8] G. O. Roberts and J. S. Rosenthal. “Optimal scaling for various metropolis-hastings algorithms”. In: **Statistical Science** 16 (2001), pp. 351–367.

[9] S. P. Brooks and A. Gelman. “General Methods for Monitoring Convergence of Iterative Simulations”. In: **Journal of Computational and Graphical Statistics** 7 (1998), pp. 434–455.

[10] J. A. Vrugt, C.J.F. ter Braak, C.G.H. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon. “Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling”. In: **International Journal of Nonlinear Sciences and Numerical Simulation** 10 (2009), pp. 273–290.

[11] 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”. In: **Water Resources Research** 44.12 (2008). W00B09, n/a–n/a. ISSN: 1944-7973.

[12] D. Vats and C. Knudson. “Revisiting the Gelman–Rubin Diagnostic”. In: **Statistical Science** 36 (2021), pp. 518–529.

DOI: 10.5540/03.2025.011.01.0376