Uncertainty quantification analysis in porous media using differential evolution MCMC method with selection (DESK)

Michel Tosin[1], Marcio R. Borges[1]

[1]*Laboratório Nacional de Computação Científica LNCC*
Av. Getulio Vargas 333, 25651-070, Petrópolis, Rio de Janeiro, Brasil
michelt@posgrad.lncc.br, mrborges@lncc.br

Abstract. Markov chain Monte Carlo methods are widely used in porous media stochastic problems. However, the sizeable stochastic dimension of the problem causes the method to have low acceptance rates. Differential evolution-based Markov chain Monte Carlo methods are good alternatives in this matter. Also, to reduce the stochastic dimension of the problem, Karhunen-Loève expansion (KLE) is commonly used to generate the permeability fields. This strategy is very effective but allows for generating fields of only one covariance function during the simulation. In that sense, variational autoencoders (VAE) come into place by generating several types of fields, leading to more realistic simulations. Then, a dataset of different fields (different covariance functions) is generated by KLE to train a VAE neural network. This work applies the Differential Evolution Markov chain Monte Carlo method with a selection step (DESK) for solving a Bayesian inference problem involving a single-phase fluid flow in a heterogeneous media. Results showed that DESK performs better than the original DE scheme. Furthermore, the VAE results are very similar to the KLE ones, demonstrating that the methodology is consistent even with a more general field generator.

Keywords: Reservoir simulation, Differential evolution MCMC with selection, Variational autoencoder

1 Introduction

Natural reservoirs exhibit high spatial variability in their hydraulic properties on multiple scales. Its diversity of scales and the lack of information (associated with the high cost of data acquisition) make the deterministic description of these properties impossible from a practical point of view and, therefore, alternative methodologies must be adopted aiming at a stochastic description of the subsurface [1, 2].

Matching field dynamic data with reservoir simulation is a stochastic inverse problem that can be formalized in Bayesian analysis and Markov chain Monte Carlo (MCMC) methods [3]. The Metropolis algorithm [4, 5] and its variants are commonly used in the Bayesian analysis of stochastic inverse problems. On that, the higher difficulty for flow problems in porous media is related to the large stochastic dimension of the fields, which leads to low acceptance rates. Several procedures have been adopted to speed up the convergence and reduce the computational cost associated with the MCMC methods. Between then, the Differential evolution (DE) MCMC proposed by Ter Braak [6] has shown great promise. Inspired by genetic algorithm ideas, the method allows the exchange of information among multiple parallel chains, which yields an appropriate scale and orientation for the jumping distribution. Tosin and Borges [7] recently presented the Differential evolution with selection mechanisms (DESK) variation of the classic DE sampler that includes a competition step to choose the candidate chains. The adaptation proved to have a higher acceptance rate and faster convergence than the original DE scheme.

Another approach often used to increase the acceptance rate of the MCMC methods consists of reducing the dimensionality of the stochastic fields. This is an effective procedure in history matching because the search is performed in a smaller parameter space. Parameterization methods such as Karhunen-Loève expansion (KLE) [8, 9] have handled the high dimensionality problem. Then, based on the available information, which is generally scarce, one must assume a specific covariance function with respective correlation lengths (or ranges). Therefore, choosing a single covariance function carries a high degree of uncertainty. On the other hand, generative neural networks are becoming very popular in replacing traditional models. The Variational autoencoder (VAE) is a generative model proposed by Kingma and Welling [10] that enforces a *prior* on the low-dimensional latent space that one can be mapped back into a realistic-looking image. Therefore, the essential characteristic of VAEs, in the context of Monte Carlo methods with Markov chains, is their ability to represent high-dimensional parametric

spaces in a low-dimensional latent space [11, 12].

This work solves a Bayesian inference problem in the scenario of a single-phase flow in a five-spot well configuration, using data on the pressure in 25 monitoring wells. To solve the problem, we use the traditional random walk (RW), DE and the DESK methods. Besides that, the authors introduce a new methodology to relax the assumption of known permeability covariance. So, more than one covariance function with different correlation lengths can be proposed. To achieve this, we also use a VAE model to generate the permeability proposals according to Xia and Zabaras [11] and Xu et al. [12]. However, unlike these works, the generative neural network is trained using sets of fields with different statistical properties (covariance functions).

The work is organized to cover our goals as follows: the present section introduces the work motivations and objectives. Section 2 describes the stochastic problem in porous media in the mathematical modeling sense. Particular attention is given to permeability modeling, which carries the stochastic element of the problem into it. Still on the random fields, Section 3 presents the KLE or VAE approaches to construct the permeability fields. In the sequel, Section 4 brings the main definitions of the Bayesian and the DESK methodology explored in the work. The results for different configurations are reunited and discussed in Section 5. Finally, Section 6 returns the main findings of the paper and indicates some natural future directions from the results.

2 Stochastic flow problem

To test the new methodology, we considered a synthetic, simplified problem: the single-phase flow in a five-spot well configuration. As seen in Figure 1, we obtain data on the pressure in 25 monitoring wells (p^{ref}) given a reference permeability field. These data are used in the likelihood function. In this work, permeability is the unknown parameter, and as we will see below, it will be treated as a random variable.

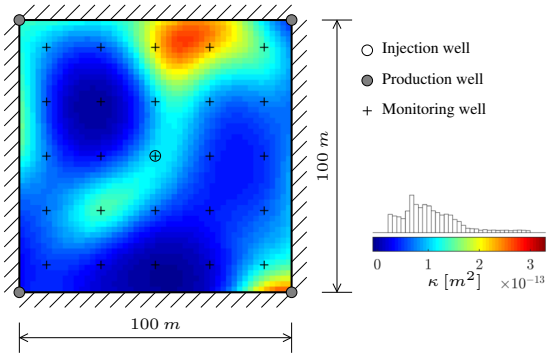


Figure 1. Domain $\mathcal{D}$ filled with the reference permeability field (κ^{ref}).

2.1 Mathematical modeling

Let $\mathcal{D} \in \mathbb{R}^2$ be the domain (with boundary Γ and unit outward normal $\mathbf{n}$) occupied by a heterogeneous and rigid porous media saturated by water. Assuming a homogeneous porosity and denoting $\mathbf{v}_{_D}$ the Darcy velocity, the single-phase flow of an incompressible fluid, in the absence of gravity, is described by the equations

$$\nabla \cdot \mathbf{v}_{_D}(\mathbf{x}) = \mathbf{q}(\mathbf{x}) \,, \quad \text{and} \quad \mathbf{v}_{_D}(\mathbf{x}) = -\frac{\boldsymbol{\kappa}(\mathbf{x})}{\mu} \nabla p(\mathbf{x}) \,, \tag{1}$$

where $\mathbf{q}$ is the source term, p and μ are the fluid's pressure and viscosity, respectively. Under the assumption of isotropy, the permeability tensor $\boldsymbol{\kappa}$ is treated here as a scalar (κ). The 2D domain has a $100 \times 100\ m^2$ dimension, discretized in a regular mesh of 50×50 elements. The permeability is peace-wise constant for each cell.

We provide homogeneous Neumann boundary conditions associated with the previous equations. A Peaceman-type model represents the wells. A flow rate ($\mathbf{q} = 100\ m^3\ day^{-1}$) is imposed on the injection well, and a constant bottom hole pressure ($p_{bh} = 101325$ Pa) on the production wells. A two-point flux approximation (TPFA) scheme approximates the solution of the mathematical problem formed by eq. (1) using the MATLAB/OCTAVE Reservoir Simulation Toolbox (MRST) simulator from Sintef [13].

2.2 Permeability modeling

Due to incomplete knowledge about the rock properties that show variability at multiple length scales, input parameters such as the permeability field, $\kappa(\mathbf{x}, \omega)$, are treated as random space functions with statistics inferred from geostatistical models (here $\mathbf{x} = (x_1, x_2)^{\mathsf{T}} \in \mathbb{R}^2$ and ω is a random element in the probability space). In line with Dagan [1] and Gelhar [2], the permeability field is modeled as a log-normally distributed function

$$\kappa(\mathbf{x}, \omega) = \beta \exp\Big[\rho \mathsf{Y}(\mathbf{x}, \omega)\Big], \tag{2}$$

where $\beta, \rho \in \mathbb{R}^+$ and $\mathsf{Y}(\mathbf{x}, \omega) \sim \mathbb{N}(\mu_{\mathsf{Y}}, \boldsymbol{\mathcal{C}}_{\mathsf{Y}})$ is a Gaussian random field characterized by its mean $\mu_{\mathsf{Y}} = \langle \mathsf{Y} \rangle$ and two-point covariance function $\boldsymbol{\mathcal{C}}_{\mathsf{Y}}(\mathbf{x}, \mathbf{y})$. Here, we consider a squared exponential covariance function

$$\boldsymbol{\mathcal{C}}_{\mathsf{Y}}(\mathbf{x}, \mathbf{y}) = \sigma_{\mathsf{Y}}^2 \exp\left(-\frac{|\mathbf{x} - \mathbf{y}|^2}{2\ell^2}\right), \tag{3}$$

with σ_{Y}^2 denoting the variance and $\ell > 0$ the correlation length. Moreover, in this work, Y is a second-order stationary process.

3 Permeability fields simulation

This subsection covers the main strategy found in the literature. However, by construction, the method depends on a fixed type of field (set by the choice of its covariance function). This implies that some permeability properties of the rock are known, which is unrealistic. This paper also proposes to relax that assumption by replacing the common generation method with a generative neural network. This approach allows for generated fields of different types during the simulation, including intermediary cases between those used in the training step. That characteristic is desired because it leads to more realistic results.

3.1 Karhunen-Loève expansion (KLE)

The Gaussian field Y can be represented as a series expansion involving a complete set of deterministic functions with correspondent random coefficients using the Karhunen-Loève (KL) expansion [8, 9] based on the eigen-decomposition of the covariance function. Depending on how fast its eigenvalues decay, one may be able to retain only a small number of terms in a truncated expansion. Consequently, this procedure may reduce the search to a smaller parameter space. Consider a random field $\mathsf{Y}(\mathbf{x}, \omega)$ defined on a probability space $(\Omega, \mathcal{A}, \mathcal{P})$ composed by the sample space, the ensemble of events and a probability measure, respectively, and indexed on a bounded domain $\mathcal{D} \in \mathbb{R}^2$. The process Y can be approximated by the truncated KLE of size m

$$\mathsf{Y}(\mathbf{x}, \omega) \approx \langle \mathsf{Y}(\mathbf{x}) \rangle + \sum_{i=1}^{\mathsf{m}} \sqrt{\lambda_i} \phi_i(\mathbf{x}) \theta_i(\omega) \,, \tag{4}$$

where λ_i is the i-th largest eigenvalue and ϕ_i its respective eigenfunction. The term $\theta_i(\omega)$ is a set of independent random variables related to λ_i and ϕ_i.

3.2 Variational autoencoders

Assume the input data set $\boldsymbol{X} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{\mathsf{N}} \left(\boldsymbol{x}^{(i)} \in \mathbb{R}^{\mathsf{N}_x}\right)$ consisting of N independent and identically distributed (*i.i.d.*) samples of the continuous (or discrete) variable drawn from the *prior* distribution $p(\boldsymbol{x})$. The idea is to represent $p(\boldsymbol{x})$ through a probabilistic model. VAEs introduces a low-dimensional latent variable $\mathbf{z} \in \mathbb{R}^{\mathsf{N}_z}$ $(\mathsf{N}_z < \mathsf{N}_x)$ to extend the representation capability of the probabilistic models [12]. To create a more treatable problem for the gradient descent methods, the framework of VAEs uses a deep neural network parametric model $\mathbb{E}_{\varphi}(\boldsymbol{x}) = q_{\varphi}(\boldsymbol{z}|\boldsymbol{x})$ (called the encoder) as an approximated probabilistic model. Thus, given a data point $\boldsymbol{x}$, it produces a distribution over the possible values of the latent value $\boldsymbol{z}$ from which the data point $\boldsymbol{x}$ could have been

generated [10]. Similarly, the decoder $\mathbb{D}_\psi(\boldsymbol{z}) = p_\psi(\boldsymbol{x}|\boldsymbol{z})$ indicates that given a latent value $\boldsymbol{z}$ it produces a distribution over the possible corresponding values of $\boldsymbol{x}$. By assuming a Gaussian encoder, with mean $\boldsymbol{\mu}$ and diagonal covariance $\boldsymbol{\sigma}$, $\boldsymbol{z}$ can be parameterized as

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\varepsilon}, \quad \text{where} \quad \boldsymbol{\varepsilon} \sim \mathbb{N}\left(\mathbf{0}, \mathbf{I}_{N_z}\right) . \tag{5}$$

The decoder is a multivariate Gaussian distribution $\mathbb{N}\left(\mathbb{D}_\psi(\boldsymbol{z}), \mathbf{I}_{N_x}\right)$.

The reconstruction loss ensures that the reconstructed image at the output is close (enough) to the input one and, here, is calculated by mean squared error (MSE). To keep the encoder outputs $\boldsymbol{z}$ close to a standard normal distribution and sufficiently diverse, we use the Kullback–Leibler divergence. ($\mathcal{D}_{\mathsf{KL}}$). Therefore, if the data points are randomly organized in batches $\boldsymbol{X}^b$, of size N_b, the total loss in a batch is given by

$$\begin{aligned}\mathcal{L}(\varphi, \psi, \boldsymbol{X}^b) &= \mathcal{L}_{\mathsf{MSE}}(\varphi, \psi, \boldsymbol{X}^b) + \mathcal{L}_{KL}(\varphi, \boldsymbol{X}^b) \\ &= \sum_{k=1}^{N_b} \left[\left|\left| \boldsymbol{x}^{(k)} - \mathbb{D}_\psi\left(\mathbb{E}_\varphi\left(\boldsymbol{x}^{(k)}\right)\right) \right|\right|^2 - \frac{1}{2}\sum_{l=1}^{N_z} 1 + \log((\boldsymbol{\sigma}^{(k)})_l^2) - (\boldsymbol{\mu}^{(k)})_l^2 - (\boldsymbol{\sigma}^{(k)})_l^2 \right] .\end{aligned} \tag{6}$$

4 Markov chain Monte Carlo method (MCMC)

The Metropolis-Hasting algorithm (MT) [4, 5] is one of the most important in history. Used to draw samples from a posterior distribution, this powerful tool is extensively applied in different areas of science. Let $\pi(\boldsymbol{\theta})$ represent the target distribution (distribution *a posteriori*) of the parameter $\boldsymbol{\theta}$. The main elements of the algorithm are the instrumental proposal distribution $\mathsf{q}(\boldsymbol{\theta}^t, \boldsymbol{\theta})$ which allows to generates new samples of $\boldsymbol{\theta}$ and the acceptation/rejection criterion α. Assuming this last one is in the classic format, the other elements will be described in this section.

4.1 Likelihood

The pressure data (measurements), denoted by p^{ref}, are combined with the *a priori* distribution ($\mathsf{P}(\boldsymbol{\theta})$) through Bayes' theorem to give

$$\pi(\boldsymbol{\theta}) = \mathsf{P}\left(\boldsymbol{\theta} \mid p^{\mathsf{ref}}\right) \propto \mathsf{P}\left(p^{\mathsf{ref}} \mid \boldsymbol{\theta}\right) \mathsf{P}(\boldsymbol{\theta}) , \tag{7}$$

that is the target (*a posteriori*) distribution of the parameter $\boldsymbol{\theta}$. Before starting the process, the *a priori* distribution reflects our best knowledge of the parameters. Assuming that the error between the reference and simulated data (p^{sim}) has a normal distribution, that is, $\mathsf{e} = |p^{\mathsf{ref}} - p^{\mathsf{sim}}| \sim \mathbb{N}\left(\mathbf{0}, \sigma^2\right)$, the likelihood is approximated as

$$\mathsf{P}\left(p^{\mathsf{ref}} \mid \boldsymbol{\theta}\right) \propto \exp\left(-\frac{|p^{\mathsf{ref}} - p^{\mathsf{sim}}|^2}{\sigma^2}\right) . \tag{8}$$

Here, σ^2 is the overall precision associated with measurement, numerical, and modeling errors. The numerator of the term within the exponential function is approximated numerically by

$$\mathsf{ER} = |p^{\mathsf{ref}} - p^{\mathsf{sim}}|^2 = \frac{\sum_{i=1}^{\mathsf{Nd}} \left[p^{\mathsf{ref}}\left(\mathbf{x}_i\right) - p^{\mathsf{sim}}\left(\mathbf{x}_i\right)\right]^2}{\sum_{i=1}^{\mathsf{Nd}} \left[p^{\mathsf{ref}}\left(\mathbf{x}_i\right)\right]^2}, \tag{9}$$

where N_d is the number of data in space.

4.2 Differential Evolution with Selection Step

The proposed distribution is an important component of the MT algorithm. Its choice must balance the capacity of variability and good convergence characteristics for the application. In this direction, [6] developed a population-based MCMC algorithm to enhance the efficiency of MCMC sampling through exchanging information among multiple parallel chains. This is called the Differential Evolution Markov chain Monte Carlo method (DE). Also inspired by genetic algorithm concepts, the Differential Evolution with Selection Step (DESK) variation of the DE method improves the acceptation rate of the algorithm by better choosing the parallel chains used during the iterations [7].

Consider the M, d-dimensional, parameters $\boldsymbol{\theta}_i^t$ ($i = 1, 2, \ldots, \mathsf{M}$) members of population $\boldsymbol{\Theta}^{(t)}$ at state t. Thus, $\boldsymbol{\Theta}^{(t)}$ is a $\mathsf{M} \times \mathsf{d}$ matrix. The expression draws the new samples

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^t + \mathsf{g}\left(\boldsymbol{\theta}_{r_1}^t - \boldsymbol{\theta}_{r_2}^t\right) + \mathbf{e}, \tag{10}$$

where $\mathbf{e} \sim \mathbb{N}\left(\mathbf{0}, b\mathbf{I}_\mathsf{d}\right)$ with b small. The factor g satisfies the optimal choice of $2.38/\sqrt{2\mathsf{d}}$ [14]. In the original DE method, $\boldsymbol{\theta}_{r_1}^t$ and $\boldsymbol{\theta}_{r_2}^t$ are randomly selected without replacement from the population $\boldsymbol{\Theta}^{(t)}/\{\boldsymbol{\theta}_i^t\}$. The DESK modification introduces a tournament selection step for candidates r_1 and r_2. A subset of size $k \geqslant 2$ elements is randomly selected from the current population ($\boldsymbol{\Theta}^{(t)}/\{\boldsymbol{\theta}_i^t\}$). Then, based on a fitness function, we selected r_1 and r_2 as the two highest-ranked individuals. Here, this function is based on the error used in the likelihood function and is defined by

$$\mathsf{f}(\boldsymbol{\theta}^t, i) \quad = \quad \frac{\xi}{|p_i^{\mathsf{ref}} - p_i^{\mathsf{sim}}|^2 + \epsilon}, \tag{11}$$

where $\xi = \max_{i \in [1,\mathsf{M}]} \left\{|p_i^{\mathsf{ref}} - p_i^{\mathsf{sim}}|^2\right\}$ and $\epsilon = 10^{-7}$ (used to avoid division by zero). The smaller the error, the larger the value of the fitness function. Additionally, when dealing with high dimensional scenarios ($\mathsf{d} \gg 1$), we propose a modification of DESK sampler (Eq. 10) inspired by the autoregressive version of Randon Walk:

$$\boldsymbol{\theta} = \left(\sqrt{1 - 2\mathsf{g}^2}\right)\boldsymbol{\theta}_{r_1}^t + \mathsf{g}\left(\boldsymbol{\theta}_{r_2}^t - \boldsymbol{\theta}_{r_3}^t\right) + \mathbf{e}. \tag{12}$$

This allows us to overcome the problem of progressive increase in the variance of the new proposals, which could cause problems in the convergence of the method. For the eq. (12) to be well defined, $\mathsf{g} = \min\left(2.38/\sqrt{2\mathsf{d}}, \sqrt{1/2}\right)$.

5 Numerical results

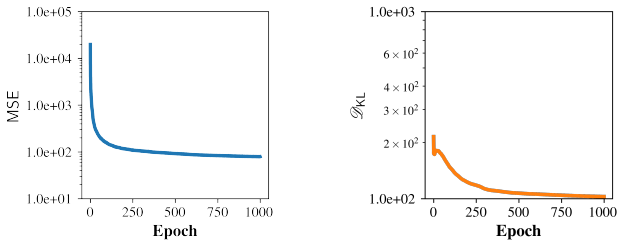
5.1 VAE training

The neural network training was performed using a dataset composed of $60{,}000$ Gaussian fields Y, where 90% of the set was explored in the training set, and the rest was split into test and validation sets. Three subsets of fields form the training set, each with squared exponential covariance and different correlation lengths ($\ell = 10, 20, 30$ at eq. (3)). The synthetic reference field was generated with $\ell = 20$ (Figure 1).

The encoder architecture comprises 4 convolutional layers and one dense layer; its latent dimension is 64. The decoder is a mirror of the encoder. A total of $1{,}000$ epochs were performed, and the data was organized into batches of 64 elements. Still about the training, the optimization process occurred through a stochastic gradient descent method, with a learning factor of 10^{-4}. The reconstruction and KL-divergency losses are shown in Figure 2. As shown, both the loss functions converged.

5.2 MCMC results

The porous media flow problem described in Section 2 with a stochastic dimension 25 was solved with the RW, DE and DESK schemes. For the last, the values of $k = 3, 5, 10, 25$ were observed. A total of $100{,}000$ iterations and 90 chains were used in all scenarios. The convergence metric applied in this work is the M-dimensional



(a) Reconstruction loss function (b) Kullback-Leibler divergence

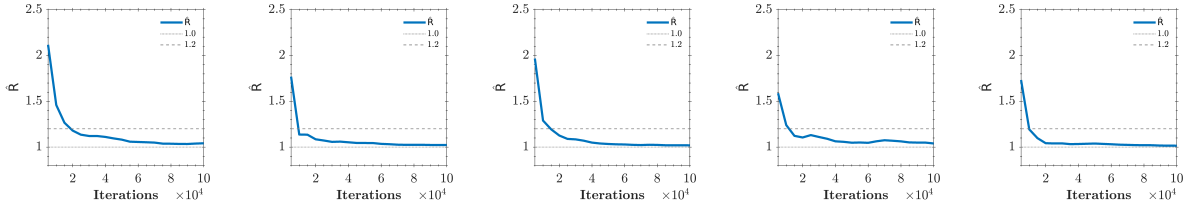
Figure 2. Convergence metrics for the VAE training. On the left is the reconstruction loss function. On the right, the $\mathcal{D}_{\mathsf{KL}}$ divergence.

multivariate potential scale reduction factor $\widehat{\mathsf{R}}$ (or MPSRF) from Brooks and Gelman [15]. In this paper, convergence is reached when the $\widehat{\mathsf{R}}$ value is stable in a region below the threshold of 1.1.

We first compare the results using KLE to generate the samples. Figure 3 presents the convergence results for the studied cases. Results (not shown here due to lack of space) reveal that all cases (except RW) have converged to the same posterior distribution. Even RW, in which we cannot declare convergence by the adopted criterion, shows that it is heading in the right direction and will eventually converge.

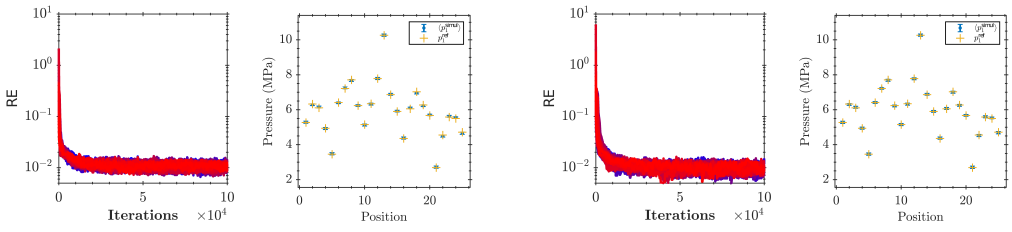
The mean acceptance rates ($\widehat{\mathsf{AR}}$) were 0.97%, 9.88%, 10.76%, 13.56%, 10.09%, and 10.10%, respectively for RW, DE, DESK$_{[3]}$, DESK$_{[5]}$, DESK$_{[10]}$ and DESK$_{[25]}$. The $\widehat{\mathsf{AR}}$ increases until $k = 5$. This is not observed for $k = 10$ and 25, which the higher selection pressure can probably explain. At first look, the gain could seem low, but it is very relevant for high-dimensional problems.

Next, we compare the results using the VAE to generate the samples. Since the trained network increased the dimension to 64 to generate fields with different correlation lengths, we compared only the RW simulations because the number of chains needed to simulate the DE scenarios properly could not be concluded in time. However, this first result already shows the high similarity between KLE and VAE schemes in the sense of posterior distribution. The VAE scheme seems to identify the correct correlation length, showing that the modified methodology with permeability fields generated by neural networks is consistent. Complementary Figure 4 gathers the graphs of error bars from the reference data with plots of relative errors of the likelihood.



(a) DE (b) DESK$_{[3]}$ (c) DESK$_{[5]}$ (d) DESK$_{[10]}$ (e) DESK$_{[25]}$

Figure 3. MPSRF convergence diagnoses results.



(a) RW (b) RW-VAE

Figure 4. Errors from eq. (9) (lefts) and reference data with mean and standard deviation of the posterior (rights).

6 Conclusions

The results show that the new DESK scheme can present better acceptance and convergence rates than the classical schemes (RW and DE) in a Bayesian inference problem applied to single-phase flow in porous media. Furthermore, we propose to use a trained neural network (VAE) to generate random samples of Gaussian fields with different correlation lengths in the Metropolis-Hastings algorithm, aiming to relax the assumption that the covariance function is fully known *a priori*. This methodology was shown to be efficient in an experiment using the RW scheme. In future work, the authors intend to extend the results by combining the DESK and VAE schemes to better conclude about the robustness of the methodology. In addition, using a VAE trained with permeability fields with more than one covariance function and several correlation lengths to generalize its application further. Another objective is to reduce the latent dimension of the autoencoder as much as possible, that is, the stochastic dimension of the problem.

Acknowledgements. The authors acknowledge the National Laboratory for Scientific Computing (LNCC/MCTI, Brazil) for providing HPC resources of the SDumont supercomputer, contributing to the research results reported in this paper. URL: http://sdumont.lncc.br. They are also grateful to Fabio Porto and Carlos Cardoso from the Data Extreme Lab research group (DEXL) for the GPU resources used to train the autoencoder. Finally, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 88887.932390/2024-00.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors or has the permission of the owners to be included here.

References

[1] G. Dagan. *Flow and transport in porous formations*. Springer-Verlag, Heidelberg, 1989.
[2] L. W. Gelhar. *Stochastic subsurface hydrology*. Englewood Cliffs. Prentice-Hall, New Jersey, 1993.
[3] 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. *Computational Geosciences*, vol. 24, pp. 1693–1706, 2020.
[4] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, vol. 21, n. 6, pp. 1087–1092, 1953.
[5] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, vol. 57, n. 1, pp. 97–109, 1970.
[6] 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 and Computing*, vol. 16, n. 3, pp. 239–249, 2006.
[7] M. Tosin and M. R. Borges. Desk-mcmc: Differential evolution markov chain monte carlo with selection mechanisms. Accepted to XLIII Brazilian National Congress on Applied and Computational Mathematics, 2024.
[8] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae, Series A*, vol. 1, 1946.
[9] M. M. Loève. *Probability Theory*. Princeton University Press, New Jersey, 1955.
[10] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International conference on learning representations*, pp. 14–27, 2014.
[11] Y. Xia and N. Zabaras. Bayesian multiscale deep generative model for the solution of high-dimensional inverse problems. *Journal of Computational Physics*, vol. 455, pp. 111008, 2022.
[12] Z. Xu, Y. Xia, and Q. Liao. A domain-decomposed vae method for bayesian inverse problems. *International Journal for Uncertainty Quantification*, vol. 14, n. 3, pp. 67–95, 2024.
[13] K.-A. Lie. *An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST)*. Cambridge University Press, Cambridge, 2019.
[14] G. O. Roberts and J. S. Rosenthal. Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, vol. 16, n. 4, pp. 351–367, 2001.
[15] S. P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, vol. 7, n. 4, pp. 434–455, 1998.