frontiers in Earth Science

ORIGINAL RESEARCH
published: 21 November 2018
doi: 10.3389/feart.2018.00203

Parallelized Adaptive Importance Sampling for Solving Inverse Problems

Christoph Jäggli, Julien Straubhaar and Philippe Renard*

Stochastic Hydrogeology and Geostatistics Group, University of Neuchâtel, Neuchâtel, Switzerland

OPEN ACCESS

Edited by:
Yuri Fialko,
University of California, San Diego,
United States

Reviewed by:
Thomas Romary,
ParisTech École Nationale Supérieure des Mines de Paris, Université de Sciences Lettres de Paris, France
Shuai Zhang,
Lawrence Livermore National Laboratory, United States Department of Energy (DOE), United States
Sarah Minson,
Earthquake Science Center, United States Geological Survey,
United States

****Correspondence:***
Christoph Jäggli
christoph.jaeggli@gmail.com

Specialty section:
This article was submitted to Solid Earth Geophysics, a section of the journal Frontiers in Earth Science

Received: *23 July 2018*
Accepted: *26 October 2018*
Published: *21 November 2018*

Citation:
Jäggli C, Straubhaar J and Renard P (2018) Parallelized Adaptive Importance Sampling for Solving Inverse Problems. Front. Earth Sci. 6:203. doi: 10.3389/feart.2018.00203

In the field of groundwater hydrology and more generally geophysics, solving inverse problems in a complex, geologically realistic, and discrete model space often requires the usage of Monte Carlo methods. In a previous paper we introduced PoPEx, a sampling strategy, able to handle such constraints efficiently. Unfortunately, the predictions suffered from a slight bias. In the present work, we propose a series of major modifications of PoPEx. The computational cost of the algorithm is reduced and the underlying uncertainty quantification is improved. Advanced machine learning techniques are combined with an adaptive importance sampling strategy to define a highly efficient and ergodic method that produces unbiased and rapidly convergent predictions. The proposed algorithm may be used for solving a broad range of inverse problems in many different fields. It only requires to obtain a forward problem solver, an inverse problem description and a conditional simulation tool that samples from the prior distribution. Furthermore, its parallel implementation scales perfectly. This means that the required computational time can be decreased almost arbitrarily, such that it is only limited by the available computing resources. The performance of the method is demonstrated using the inversion of a synthetic tracer test problem in an alluvial aquifer. The prior geological knowledge is modeled using multiple-point statistics. The problem consists of the identification of $2 \cdot 10^4$ parameters corresponding to 4 geological facies values. It is used to show empirically the convergence of the PoPEx method.

Keywords: adaptive importance sampling, machine learning, uncertainty quantification, bayesian inversion, monte carlo, multiple-point statistics, parallelization

1. INTRODUCTION

Inverse problems play a key role in almost all the geosciences. Indeed, this is often the only approach allowing to identify hidden structures of the interior of the earth and to estimate the physical properties of the buried rocks from indirect physical measurements at the surface or in a few boreholes. In groundwater hydrology, the aim is generally to infer the position of highly permeable or impermeable rocks and estimate their porosities and permeabilities from punctual measurements of state variables (e.g., hydraulic heads, tracer concentrations, water temperature, etc.). As for any geophysical problem, inverse methods are of utmost importance and a fundamental step in most quantitative hydrogeological studies (de Marsily et al., 2000; Carrera et al., 2005; Zhou et al., 2014) as well as many environmental modeling problems (Moles et al., 2003; Wainwright and Mulligan, 2005).

However, despite its huge significance and despite more than 50 years of research on this topic in geophysics and hydrology, current methods are still unable to solve certain types of problems efficiently. For instance, an open problem is to solve probabilistic inverse problems that involve discrete structures such as channels, lenses, karst conduits, or faults which cannot be represented by standard multi-Gaussian fields (Gómez-Hernández and Wen, 1998; Journel and Zhang, 2006). The identification and representation of such geological features is indispensable because it heavily controls fluid flow in the underground (Feyen and Caers, 2006). Using a wrong and smoothed representation of such discrete features is known to bias significantly the groundwater forecasts and corresponding uncertainty analysis (Gómez-Hernández and Wen, 1998; Kerrou et al., 2008).

To overcome this difficulty, different approaches have been developed and were recently reviewed by Linde et al. (2015). One general strategy is to construct first a probabilistic prior able to represent stochastic but geologically realistic structures and to embed it in the inverse method. Often, this geological prior can take only discrete values representing the rock types or some specific geological features.

Inverse methods relying heavily on continuity assumptions or simple statistical distributions (typically multi-Gaussian) are not capable to manage this type of problems. On the opposite, sampling algorithms can account for such complex setup (Oliver et al., 1997; Robert and Casella, 2004; Fu and Gómez-Hernández, 2008; Mariethoz et al., 2010a; Hansen et al., 2012; Laloy et al., 2016; Rubinstein and Kroese, 2016). These methods represent the solution of the inverse problem as a set of models (or samples) describing the posterior distribution. From this set of samples, one may approximate any quantity of interest such as mean values, maximum likelihood values, uncertainty bounds, or probabilities of characteristic events. Unfortunately, for most of these approaches, the computational effort is extremely demanding (Fu and Gómez-Hernández, 2008; Romary, 2010; Linde et al., 2015) and the challenge is to design an efficient sampling scheme able to deal with categorical information in the prior distribution.

In a previous paper (Jäggli et al., 2017), we proposed the Posterior Population Expansion (PoPEx) algorithm to expand iteratively an existing set of geological models. PoPEx was specifically designed for handling discrete parameter values, even if it can be applied to the continuous case as well. The discrete parameter fields can be generated with any geostatistical method.

In our previous paper and in this one, we use a multiple-point statistics technique for expressing the prior distribution because this allows the user of PoPEx to formulate its prior geological knowledge in the area where he is carrying out the inversion. This knowledge is expressed by providing a training image (TI). Multiple-point statistics (MPS) simulation techniques (Strebelle, 2002; Arpat and Caers, 2007; Honarkhah and Caers, 2010; Mariethoz et al., 2010b; Straubhaar et al., 2013) can learn the spatial patterns from the TI and can produce stochastic simulations that resemble the TI. The simulations can be conditioned by local values if they are known (hard data). The advantage of that approach is that it is flexible. The same code can generate all kind of geological structures (channels, lobes, braided systems, fractures, etc.) and therefore it can be applied to a very wide range of inverse problems and applications.

Like most sampling techniques, PoPEx produces iteratively new parameter fields (the samples) using a geostatistical technique (see for example the book of Chilès and Delfiner, 2009), then runs the forward problem (in our case a groundwater flow and transport simulation, but it could be any forward operator), evaluates the misfit and likelihood for that solution, and accumulates novel knowledge. At each iteration, the geostatistical simulation algorithm is controlled by PoPEx: the general mechanism is to condition the simulation of the parameter fields with a set of punctual values (hard data) selected preferentially from previous models having a high likelihood.

This method proved to be very efficient on a synthetic example (Jäggli et al., 2017): a comparison with two existing Markov chain Monte Carlo (McMC) methods showed that the method was able to considerably decrease the computational cost. But this study also allowed us to identify that the initial version of PoPEx produced slightly biased predictions.

In this paper, we revisit completely the core of the PoPEx algorithm. The overall goal is to improve the usability, accuracy and computational time. The most important contribution is to introduce a new strategy allowing to produce unbiased predictions. The bias happens because the generation of a new realization is influenced by all the previous models in the chain. This sampling strategy favors some realizations over others. When computing predictions, however, these correlations must be taken into account. In other words, we propose to consider the method as an *adaptive importance sampling (AIS)* (Naylor and Smith, 1988; Oh and Berger, 1992; Murphy, 2012) and suggest a simple technique to produce unbiased predictions. The additional computational cost is negligible and does not increase the overall running time. From this perspective, the method can be interpreted as an unsupervised machine learning scheme that aims to learn an optimal probability density which can be used in the AIS scheme. The class of inverse problems that can be addressed is very broad and goes beyond applications in the field of geostatistics. The only requirements are a forward problem solver, an inverse problem description (including the likelihood function), and a conditional simulation tool (e.g., any geostatistical method) that generates models according to the prior distribution.

On top of that, we show how the algorithm, together with all modifications, can be parallelized. We show that it scales perfectly in the considered example. Hence, the computational time is directly reduced by the number of parallel chains, without compromising the outcomes. This is a powerful result, because the main hindrance against the use of sampling strategies is the computational costs. With the proposed methodology, models can be produced in parallel. The only limitations concerns the number of available CPU's, or more precisely, the number of forward problem evaluations that can be run in parallel. Today, most research and engineering groups have access to high performance computer facilities, and therefore these requirements are not too restrictive.

The paper is organized as follows. Section 2 provides the required background related to the inverse problem and the general concepts of the method before explaining the details of the modified algorithm. A case study together with a convergence analysis is presented in section 3. Finally, in section 4, the advantages and limitations of the methodology are discussed and summarized.

2. METHODOLOGY

In this section, we first review the general definition of the inverse problem following the notations and approach from Tarantola (2005). Then we introduce the most important techniques constituting the base of PoPEx (Jäggli et al., 2017). As a consequence, the first part of this section mainly presents material that has been proposed and discussed elsewhere. It is toward the end of section 2.2 and in section 2.3 that we present the novel methods that constitute the core of this paper.

2.1. Inverse Problem

The general inverse theory presented by Mosegaard and Tarantola (2002) and Tarantola (2005) contains the commonly used Bayesian formulation as special case. Furthermore, it lives without the (problematic) notion of conditional probabilities (e.g., Borel's paradox) and alternatively uses the concept of *states of information*. In the following, we slightly enrich their explanations with a few comments specifically dedicated to the hydrogeological framework.

Solving an inverse problem is usually related to honoring a sparse set of observations $\mathbf{d}^{\text{obs}} = \{d_1^{\text{obs}}, \ldots, d_m^{\text{obs}}\}$ called **data**. The nature of these observations can differ widely and may depend on the overall framework. When studying subsurface properties, they often represent measurements of state variables such as hydraulic heads, production data or contaminant concentration. Due to imperfect measuring devices, these quantities usually include uncertainties. It is common to use a finite set of parameters $\mathbf{m} = \{m_1, \ldots, m_n\}$ to fully describe the physical system under study. Any possible collection of such values will henceforth be called a **model** or equivalently a **realization**. In this regard, a model can cover a vast number of physical and conceptual quantities, as, for instance, boundary conditions, hydraulic conductivity maps, or specific storage values. The collection of all possible models is called **model space** and is denoted by $\mathcal{M}$. In the hydrogeological framework, a common approach is to subdivide an aquifer into a finite number of volume elements (simulation grid) and characterize the hydraulic conductivity in each grid cell. In this case, the underlying model $\mathbf{m}$ includes one parameter m_i per grid element, that defines the physical property in this small sub-domain. The choice of a set of representative dimensions is equivalent to the definition of a parametrization of $\mathcal{M}$. Note that for a given system, such a coordinate system is not unique. "Permeability," for example, can be replaced by "resistivity," "speed" with "slowness" or "frequency" with "period."

In practice it is possible to observe parameters that can also be included in $\mathbf{m}$. Boreholes, for example, often provide cores, from which petrophysical values can be deduced with high precision. If the model space is designed to describe the same quantities, we simply remove the corresponding degrees of freedom from any possible model $\mathbf{m}$, and reduce the number of dimensions in the model space $\mathcal{M}$.

In many fields, well-founded physical theories have been established in order to describe processes and interactions. They can be used to describe relations between the models and the observations. From a naïve point of view, it means that for a given model $\mathbf{m}$ the error-free values of the corresponding data set $\mathbf{d}$ can be predicted. This theoretical link between a model and the observable parameters is called the forward problem and described by $\mathbf{d} = \mathbf{g}(\mathbf{m})$. The function $\mathbf{g} = \{g_1, \ldots, g_m\}$ denotes the **forward operator**. Tarantola (2005) formulated the probabilistic solution of an inverse problem as a non-negative measure function that combines two different states of information. Typically, these states of information are captured by the prior and the likelihood function. The prior distribution $\rho(\mathbf{m})$ describes any available information on the model parameters, that is independent of the data set. The likelihood function, $L(\mathbf{m})$, usually embeds the forward operator and is a probabilistic measure of how well a given model is able to explain the observations. The solution, called the **posterior distribution**, of an inverse problem is the conjunction of the prior and the likelihood operator such as

$$\sigma(\mathbf{m}) = c\,\rho(\mathbf{m})L(\mathbf{m}), \tag{1}$$

where c is a normalization constant. In the Bayesian framework, the posterior measure is considered to be the product of (conditional) probability distributions. The latter approach is contained in Equation (1) and applies under some regularity conditions. For this reasons, the formulation by Tarantola (2005) is more general.

2.2. Posterior Population Expansion (PoPEx)

It is worthwhile to recall several important concepts, that originally have been introduced by Jäggli et al. (2017). Afterwards, some small improvements will be suggested. These modifications just slightly influence the evolution of the sampling scheme, so that we decided not to rename the method and still call it Posterior Population Expansion (PoPEx). The general approach of the PoPEx algorithm is to generate a large number of models $\mathbf{m}_1, \ldots, \mathbf{m}_N$ that represent the posterior probability density in Equation 1. From this approximation it is possible to compute posterior probabilities of events. The sampling procedure, however, requests to compute $\sigma(\mathbf{m_k})$ for every $k = 1, \ldots, N$, what can be highly intensive in terms of computational costs. For this reason, the main idea of the PoPEx method is to make the sampling as efficient as possible. Each generation of a new model $\mathbf{m}_k$ is therefore guided by all the previous samples $\mathbf{m}_1, \ldots, \mathbf{m}_{k-1}$. For doing so, information maps (denoted by P^k and $D(P^k||Q)$, see below) are computed iteratively and ensure that the sampling of $\mathbf{m}_k$ is strongly guided by 'good' models with high posterior values. The transfer of information from $\mathbf{m}_1, \ldots, \mathbf{m}_{k-1}$ to $\mathbf{m}_k$ runs through a set of value restrictions imposed on the new model (denoted by HD^k, see below).

2.2.1. Set of Models $\mathcal{M}^k$

The underlying algorithm is able to examine many different types of uncertainties and parameter identification problems. It is possible, for example, to consider parameters concerning boundary and/or initial conditions, spatial heterogeneities, recharge time series, etc. The model set $\mathbf{m}$ is then simply subdivided into different parts $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \dots\}$, where each $\mathbf{m}_i = \{m_{i_1}, \dots, m_{i_r}\}$ represents one specific parameter type. The only requirement is that samples representing that uncertainty can be generated from a *conditional simulation tool.*

In order to keep the following descriptions as simple as possible, we will only consider one type of model parameters and write $\mathbf{m} = \{\mathbf{m}_1\} = \{m_1, \dots, m_n\}$. This set will be used to describe spatial heterogeneities of hydraulic permeabilities and is generated by a pixel based MPS technique (Strebelle, 2002; Mariethoz et al., 2010b; Straubhaar et al., 2011). Such methods require a spatial subdivision of the computational domain into a finite number of $n \in \mathbb{N}$ elements (**pixels**). The union of all pixels is called the **simulation grid**. MPS generate realizations of a random variable by reproducing multiple-point statistics from a training image. Each realization can be associated to a model $\mathbf{m} = \{m_1, \dots, m_n\}$ by putting the MPS value from pixel j into the parameter m_j. In the example above, a variable m_j could then be linked to the constant permeability (or resistivity) in the j-th volume element of the computational domain. The term "linked" is used because it is not uncommon for the model parameters m_j to not contain permeability (or resistivity) values directly but only conceptual representatives of such. For the present work, it is assumed that the prior probability density ρ is precisely the distribution of the MPS random variable. Therefore, using the MPS machine to produce independent and unconditioned models is equivalent to drawing realizations from ρ. It is important to note that conditioning simulators work sequentially. This means that they start by randomly selecting a permutation ς over the set of indices $\{1, \dots, n\}$ that defines the order in which the components of a new model are treated. Whenever $m_{\varsigma(j)}$ is about to get informed, conditional simulation tools only consider previously simulated components and draw $m_{\varsigma(j)}$ according to the probability

$$\mathbb{P}(\,\cdot \mid m_{\varsigma(1)}, \dots, m_{\varsigma(j-1)}).$$

In other words, at this point of the simulation, $m_{\varsigma(j)}$ is considered to be independent of any uninformed component in $\mathbf{m}$.

Sampling a model space for solving an inverse problem, means to iteratively produce a finite number of N realizations

$$\mathbf{m}_1 \to \mathbf{m}_2 \to \cdots \to \mathbf{m}_N,$$

that characterize (in some way) the posterior distribution. During this procedure, the likelihood function must be evaluated for every model in the chain. It is not uncommon that this computation is very demanding and represents the most important source of computational cost. After each iteration $k = 1, \dots, N$, the models can be assembled within the collection

$$\mathcal{M}^k = \{\mathbf{m}_1, \dots, \mathbf{m}_k\}, \tag{2}$$

while the normalized likelihood values

$$\tilde{L}(\mathbf{m}_j) = \frac{L(\mathbf{m}_j)}{\sum_{r=1}^{k} L(\mathbf{m}_r)}, \qquad j = 1, \dots, k,$$

are joined in $\tilde{L}^k = \{\tilde{L}(\mathbf{m}_1), \dots, \tilde{L}(\mathbf{m}_k)\}$. The tilde notation indicates that a normalization has been applied, a convention that will be used throughout this paper. There are two different kinds of normalization that will be used. In the latter equation, the total weight was computed by summing all likelihood values from the previous iterations. This action must be renewed, whenever a new model $\mathbf{m}_{k+1}$ is sampled. Secondly, we will define spatial maps. The normalization is then performed through all locational values, and the resulting map can be interpreted as a spatial probability density (c.f. Equation 5).

2.2.2. Probability Maps Q and P^k

The possible value range for each model parameter m_i depends on the TI. After defining a set of $s - 1$ threshold levels this range may be separated into s different categories, called **facies values** or simply **facies** and denoted by $\{f_1, \dots, f_s\}$. When working with discrete models, these categories usually define a one-to-one relation to the set of all possible values in the TI. From the facies values, it is possible to establish a collection of pixel-based indicator functions. If $\mathbf{m}$ is a given model and each pixel $j \in \{1, \dots, n\}$ is represented by its center location $\mathbf{x}_j$, these functions are defined as

$$\mathbf{1}_{f_i}(\mathbf{m}; \mathbf{x}_j) = \begin{cases} 1 & \text{if } m_j \text{ belongs to category } f_i \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Any linear combination of the quantities in Equation (3) can be interpreted as a map with constant value in each pixel. The concept of these indicator functions is very important throughout the present paper. If the precise pixel location $\mathbf{x}_j$ is not relevant, we will henceforth omit its explicit notation. The indicator functions help to compute moments of the random vector that is associated to the MPS tool. Let q_i represent the pixel-wise probability of the model values to fall into category f_i. If $\mathbb{E}(\cdot)$ denotes the usual expectation operator, they read

$$q_i = \mathbb{E}\big(\mathbf{1}_{f_i}(\mathbf{m})\big), \quad i = 1, \dots, s.$$

The set $Q = \{q_1, \dots, q_s\}$ then collects all the *prior probability maps* for the facies categories. If the MPS machine is trained to produce stationary and unconditioned simulations, then the maps q_i are constant over the computational domain and equal the corresponding facies proportion in the training image. On the other hand, a set $\mathcal{M}^k = \{\mathbf{m}_1, \dots, \mathbf{m}_k\}$ can be used to define a second collection $P^k = \{p_1^k, \dots, p_s^k\}$ such that

$$p_i^k = \sum_{j=1}^{k} \mathbf{1}_{f_i}(\mathbf{m}_j)\tilde{L}(\mathbf{m}_j). \tag{4}$$

The superscript k in the notation p_i^k indicates the number of realizations that has been used in its computation. It is important to perceive the consequences of weighting the summands by the normalized likelihood values $\tilde{L}(\mathbf{m}_j)$. If $\mathbf{m}_{j_0}$ is a model with a large likelihood value (with respect to the other ones), this means that some facies patterns in $\mathbf{m}_{j_0}$ may be very important. Therefore, the probability maps in Equation (4) are formed by weighting "good" facies patterns more heavily than "bad" ones. Consequently, these maps may be able to provide information that can be used to generate "good" models. But at this point it is unclear where this information can be found and how it could be used. The answer to this question lies in the relation between Q and P^k. The central idea of the PoPEx sampling is to consider and learn from all models $\mathbf{m}_1, \ldots, \mathbf{m}_k$, before generating $\mathbf{m}_{k+1}$. This procedure can be split into two parts, that will be explained in the following.

2.2.3. Kullback-Leibler Divergence $D(P^k||Q)$

Kullback and Leibler (1951) introduced a measure called Kullback-Leibler divergence (KLD) to compare two probability distributions. It computes how a candidate probability diverges from an expected one. This is precisely what is needed to measure the information content of P^k with respect to Q. In other words, the Kullback-Leibler divergence can be used to identify pixel locations, where the facies probabilities in P^k are "extreme" with respect to Q. It is given by

$$D(P^k||Q) = \sum_{i=1}^{s} p_i^k \log\left(\frac{p_i^k}{q_i}\right). \tag{5}$$

Whenever $q_i > 0$ for all $i = 1, \ldots, s$, this equation is well defined. But let's assume that there is $i \in \{1, \ldots, s\}$ and a pixel $\mathbf{x}_j$ with $q_i(\mathbf{x}_j) = 0$. This means that it is impossible for the MPS tool to produce a model $\mathbf{m}$ where the value m_j falls into the i-th category. From Equation (4) it follows that $p_i^k(\mathbf{x}_j)$ must vanish as well. In short, $q_i(\mathbf{x}_j) = 0$ implies $p_i^k(\mathbf{x}_j) = 0$, and the corresponding terms in Equation (5) can be ignored. A brief comment on the prior maps q_i may help to enhance the meaning of Equation (5). If there is a large set of independent models $\{\mathbf{m}_1, \ldots, \mathbf{m}_N\}$ that is distributed according to ρ, the law of large numbers (LLN) [c.f. Durrett (2010)] suggests to use approximations

$$q_i \approx \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}_{f_i}(\mathbf{m}_j). \tag{6}$$

From this perspective, the relation between p_i^k and q_i is easier to detect. Both definitions use the same indicator functions, but are weighted differently. $D(P^k||K)$ provides a pixel based information map, that indicates how surprising the facies patterns become, whenever they are weighted by the likelihood values. As mentioned earlier, it is possible to normalize the Kullback-Leibler divergence map spatially. The rescaled map is denoted by $\tilde{D}(P^k||Q)$ and can be interpreted as a discrete probability density defined over the pixel locations.

2.2.4. Hard Conditioning Data HD^k

We mentioned earlier, that each model must be generated by a "conditional simulation tool." This means that it must be possible to condition (impose) some of the values in $\mathbf{m}$. Doing so allows fields that honor local data, commonly known as **hard conditioning (HD)** (Mariethoz and Caers, 2014), to be generated. The enforced value v together with the pixel location $\mathbf{x}$ forms one conditioning object, denoted by $(\mathbf{x}, v)$. A reliable set of hard conditioning data may enhance the chance to generate a new model $\mathbf{m}_{k+1}$ that provides a large likelihood value $L(\mathbf{m}_{k+1})$. Considering the previous explanations, it seems natural to sample a set $\{x_1, \ldots, x_{n_k}\}$ of hard conditioning locations (*where* conditioning should apply) from the normalized Kullback-Leibler information $\tilde{D}(P^k||K)$. For every selected position $\mathbf{x}_i$, we can then sample a model index $j \in \{1, \ldots, k\}$ according to $\tilde{L}^k$ and extract the conditioning value (*which* value should be imposed) from $\mathbf{m}_j(\mathbf{x}_i)$. This produces a set of hard conditioning data $HD^k = \{(\mathbf{x}_1, v_1), \ldots, (\mathbf{x}_{n_k}, v_{n_k})\}$.

So far, nothing original has been proposed. The modifications that we suggest now, concern the number of elements in HD^k. Jäggli et al. (2017) started with a set of unconditioned models, before fixing the number of conditioning points to a user defined parameter and leaving it unchanged. However, the statistical significance and robustness of the algorithm could certainly be increased by adding some "randomness" into this selection procedure. We suggest to change randomly the number of conditioning points in each iteration. For this, we suggest to fix an upper bound $n_{\max}$, and draw the number of conditioning points from an uniform distribution over the set $\{0, 1, \ldots, n_{\max}\}$. The amount of hard conditioning data n_k thus may change in each iteration k. It is therefore possible to occasionally generate unconditioned realizations.

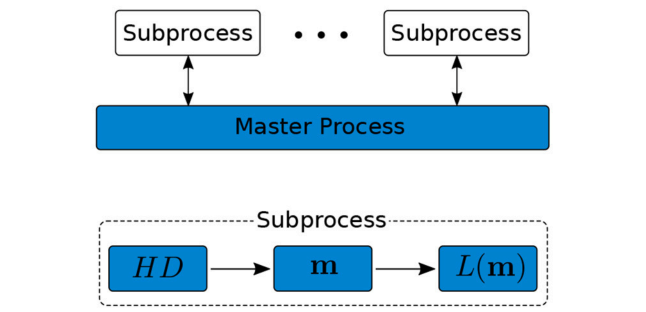


FIGURE 1 | Overview of the parallelized PoPEx procedure.

2.2.5. Parallelization of the Algorithm

Every loop of the PoPEx algorithm consists in four main steps: derive a set of hard conditioning points, generate a new model, compute its likelihood value and update the Kullback-Leibler divergence map. One strategy to parallelize this procedure is to encapsulate the first three steps in a subprocess separated from the last one. Then, a master process launches such subprocesses in parallel on other CPU's. Each subprocess is simply fed by the current available KLD map and performs the enclosed steps independently. After the result of a subprocess is communicated back to the master process, this latter updates the KLD map and launches another subprocess. A brief overview of this workflow is presented in **Figure 1**.

The pseudocodes of the parallelized PoPEx algorithm and the corresponding subprocesses are given in the algorithms 1 and 2, respectively. The variable "manager" appearing within the main algorithm is a FIFO ("first in first out") queue of maximal length n_{par} that maintains the communication toward the subprocesses. FIFO stands for queues where new elements are appended at the tail (line 9) and removed from its head (line 11). In this regard, the lines 5-10 of algorithm 1 are designed to launch n_{par} parallel subprocesses (line 8) and retain corresponding handles (line 9). The lines 11-15 on the other hand, check the status of the first subprocess (line 12) and react accordingly. If it has terminated, their outputs are received (line 13) and the corresponding variables are updated (line 14). If it is still running however, the handle is sent to the back of the queue (line 17). The main motivation for appending the running subprocesses at the end is to rapidly detect and remove other jobs that have been completed. But as a consequence, reproducibility of the algorithm is not guaranteed. If reproducibility is crucial, we could simply change line 17 such that the processes are re-appended at the head of the queue and ensure that the first n_{par} workers are launched before lines 11-18 may apply.

Algorithm 1 PoPEx

```
1:  **Input:** $n_{\max}$, $n_{\text{par}}$, $N$ and $Q$
2:  $k \leftarrow 0$ and $P^0 \leftarrow Q$
3:  manager ← empty queue          # FIFO queue
4:  **while** $k < N$ **do**
5:    $n_m$ = length(manager)
6:    **if** $n_m < n_{\text{par}}$ **and** $k + n_m < N$ **then**
7:      p ← **new** subprocess
8:      p.start(Subprocess($\mathcal{M}^k, \tilde{L}^k, D(P^k||Q), n_{\max}$))
9:      manager.append(p)
10:   **end if**
11:   p ← manager.pop()
12:   **if** p.ready() **then**
13:     $(\mathbf{m}_{k+1}, L(\mathbf{m}_{k+1}))$ = p.get()
14:     update $\mathcal{M}^k$, $\tilde{L}^k$ and $D(P^k||Q)$
15:     $k \leftarrow k + 1$
16:   **else**
17:     manager.append(p)
18:   **end if**
19: **end while**
```

Algorithm 2 Subprocess

```
1: **Input:** $\mathcal{M}^k$, $\tilde{L}^k$, $D(P^k||Q)$ and $n_{\max}$
2: **Output:** $\mathbf{m}_{k+1}$ and $L(\mathbf{m}_{k+1})$
3: sample $n_k \sim U(0, n_{\max})$
4: $HD^k \leftarrow \text{hd}(n_k, \mathcal{M}^k, \tilde{L}^k, D(P^k||Q))$
5: $\mathbf{m}_{k+1} \leftarrow \text{model}(HD^k)$
6: $L(\mathbf{m}_{k+1}) \leftarrow \text{likelihood}(\mathbf{m}_{k+1})$
```

Calling "$\text{hd}(n_k, \mathcal{M}^k, \tilde{L}^k, D(P^k||Q))$" within a subprocess (algorithm 2, line 4), uses the above strategy to compute a set of n_k hard conditioning couples. On the other hand, the methods "model(HD^k)" (line 5) and "likelihood($\mathbf{m}$)" (line 6) are application dependent functions that generate a new model from a given set of conditioning data and compute the corresponding likelihood value.

In practice it might be unclear how to provide a suitable collection Q. Assuming that the involved modeling tool samples from the prior distribution, opens the door to approximate Q. Even before launching the PoPEx algorithm, we could produce a sufficiently large number of unconditioned models, and approximate Q by Equation (6). As the effort of generating a model is often negligible with respect to the computation of the likelihood value, the additional cost for approximating Q is unimportant. If there is a considerable effort required to generate a model, we could also consider to start with an initial guess of Q and iteratively improve it. However, changing Q along the sampling procedure may render the algorithm unstable.

2.3. Posterior Prediction of Events

Solving an inverse problem, not only serves to represent the posterior measure function, but also aims to compute the (posterior) probability of events $A \subset \mathcal{M}$. More generally, we would like to compute integrals with respect to σ, such as

$$\mu = \int_{\mathcal{M}} f(\mathbf{m}) d\sigma, \tag{7}$$

where $f(\cdot)$ is an operator that expresses some quantity of interest. Because the model space $\mathcal{M}$ and the posterior measure function σ can be very complex, an analytical solution of these integrals is usually not available. The generic term importance sampling (IS) (Hesterberg, 2003; Robert and Casella, 2004; Liu, 2008; Rubinstein and Kroese, 2016) stands for a framework that provides approximations of such integrals by a weighted sum over a large number of realizations. Because it is often difficult or inefficient to directly sample from the distribution σ, importance sampling suggests instead, to draw realizations from a sampling distribution ϕ and weight the summands proportionally to the ratio $\sigma(\mathbf{m})/\phi(\mathbf{m})$. To find and use an appropriate sampling distribution ϕ however, can be challenging.

We propose to consider the PoPEx algorithm as a procedure, that iteratively learns and adapts the sampling distribution ϕ_k. During this procedure, all the previously generated realizations are combined and used to localize important regions in the model space. This is known as adaptive importance sampling (AIS) and has been introduced in a econometric framework (Naylor and Smith, 1988; Oh and Berger, 1992). The generation of a new model $\mathbf{m}_{k+1}$ is understood as to randomly draw one sample according to ϕ_k. By construction, this distribution must include the random selection of HD^k as well as the conditional modeling tool. For each model in a chain of N realizations, we compute a weight ratio $w_k = \sigma(\mathbf{m}_k)/\phi_k(\mathbf{m}_k)$ and estimate the integral μ by

$$\hat{\mu} = \sum_{k=1}^{N} f(\mathbf{m}_k)\tilde{w}_k. \tag{8}$$

Again, the tilde notation was used to indicate normalized weights $\tilde{w}_k$ such that

$$\tilde{w}_k = \frac{w_k}{\sum_j w_j}.$$

Several remarks are worth being considered. The computation of Equation (8) only uses normalized weights. Therefore, it is not required to know precisely the normalization constant of either σ or ϕ. Furthermore, the computation of the weights w_k can be simplified by using the factorization of σ (c.f. Equation 1). Each likelihood value $L(\mathbf{m}_k)$ is evaluated during the PoPEx procedure, so that for constructing the weights, it is sufficient to compute the ratio

$$\frac{\rho(\mathbf{m}_k)}{\phi_k(\mathbf{m}_k)}, \quad \text{for } k = 1, \dots, N. \tag{9}$$

Roughly speaking, this ratio compares the probability measure of generating a model $\mathbf{m}_k$ with and without observed data.

Computation of the Sampling Weights

In every iteration of the PoPEx algorithm, a set of location-value pairs is derived and imposed as hard conditioning for the next model. We will show in this section, that when using a pixel-based MPS technique to generate the models, the sampling ratios in Equation (9) only depend on the sets HD^k. Let us consider $HD^k = \{(\mathbf{x}_1, v_1), \dots, (\mathbf{x}_{n_k}, v_{n_k})\}$ and distinguish two events that henceforth will be noted similarly:

1. The MPS scheme is assumed to follow the prior probability measure ρ. Recall that such a tool iteratively supplies pixels with simulation values from a training image. HD^k appearing within ρ will refer to the event where "the first n_k locations (met during the simulation) were $\{\mathbf{x}_1, \dots, \mathbf{x}_{n_k}\}$ and they obtained the values $\{v_1, \dots, v_{n_k}\}$." Following this line, $\rho(\mathbf{m}|HD^k)$ expresses the conditional measure to draw $\mathbf{m}$, when the first n_k assignments imposed the values $\{v_1, \dots, v_{n_k}\}$ at the locations $\{\mathbf{x}_1, \dots, \mathbf{x}_{n_k}\}$.
2. The sampling distribution on the other hand takes the PoPEx iterations into account. HD^k appears within ϕ_k whenever we want to indicate that "during the k-th PoPEx iteration, HD^k has been produced and used as hard conditioning." Accordingly, $\phi_k(\mathbf{m}|HD^k)$ measures the probability of sampling $\mathbf{m}$ at iteration k, knowing that HD^k has been imposed.

Henceforth, we will only consider combinations $\mathbf{m}$ and HD^k that produce strictly positive measure values (i.e., where the values on the n_k locations coincide). Furthermore, we will assume that all the conditioning binomials in HD^k are independent of each other. This assumption is reasonable if the conditioning locations are well separated. It is therefore necessary, that the number of conditioning points is adequate with respect to the simulation grid. The MPS processes involved behind $\rho(\mathbf{m}|HD^k)$ and $\phi_k(\mathbf{m}|HD^k)$ are the same. It follows that the two measure values must be equal, and thus

$$\frac{\rho(\mathbf{m})}{\phi_k(\mathbf{m})} = \frac{\rho(\mathbf{m})}{\rho(\mathbf{m}|HD^k)} \frac{\phi_k(\mathbf{m}|HD^k)}{\phi_k(\mathbf{m})}.$$

Using the definition of conditional probabilities, the ratio can be rearranged as

$$\frac{\rho(\mathbf{m})}{\phi_k(\mathbf{m})} = \frac{\rho(HD^k)}{\rho(HD^k|\mathbf{m})} \frac{\phi_k(HD^k|\mathbf{m})}{\phi_k(HD^k)}.$$

Standard techniques from the field of combinatorial probability allow to express all the above quantities. On the one hand, $\rho(HD^k|\mathbf{m})$ measures the probability of informing the first n_k pixels according to HD^k, when the sampled model is known. But knowing $\mathbf{m}$ implies that the conditioning values in HD^k are given, so that we only need to compute the probability to meet the n_k conditioning locations (in any order) in the very beginning of the MPS simulation. If there are n pixels in the simulation grid, $\rho(HD^k|\mathbf{m})$ is given by

$$\rho(HD^k|\mathbf{m}) = \frac{n_k!(n-n_k)!}{n!}.$$

On the other hand, because the hard conditioning data is assumed to be independent, $\rho(HD^k)$ reads

$$\rho(HD^k) = \frac{n_k!(n-n_k)!}{n!} \prod_{j=1}^{n_k} \rho(v_j; \mathbf{x}_j),$$

where $\rho(v_j; \mathbf{x}_j)$ is the prior probability of meeting the value v_j at location $\mathbf{x}_j$. In section 2.2, the simulation values have been categorized into the set $\{f_1, \dots, f_s\}$. For a fixed HD^k, let us define an index-to-index map $r = r(j)$ such that $f_{r(j)}$ identifies the category of v_j. An approximation to $\rho(v_j; \mathbf{x}_j)$ can be obtained from Equation (6) by specifying $\rho(v_j; \mathbf{x}_j) \approx q_{r(j)}(\mathbf{x}_j)$. This simply suggests to find the map $q_{r(j)}$ that corresponds to the category of v_j and extract the probability value at $\mathbf{x}_j$.

Every iteration contains the following three steps. Select a number n_k, sample conditioning locations from $\widetilde{D}(P^k||Q)$ and extract conditioning values by weighting the simulations according to the computed likelihood measures in $\tilde{L}^k$. They are performed independently such that the probability of selecting HD^k, knowing $\mathbf{m}$, is measured as

$$\phi_k(HD^k|\mathbf{m}) = \phi_k(n_k) \prod_{j=1}^{n_k} \widetilde{D}(P^k||Q)(\mathbf{x}_j)$$

while similarly (with the hard conditioning data points being independent)

$$\phi_k(HD^k) = \phi_k(n_k) \prod_{j=1}^{n_k} \phi_k(v_j; \mathbf{x}_j) \widetilde{D}(P^k||Q)(\mathbf{x}_j).$$

The value $\phi_k(n_k)$ is the probability of selecting n_k while the measure $\phi_k(v_j; \mathbf{x}_j)$ is the probability to draw a model (according to $\tilde{L}^k$) that presents the value v_j at location $\mathbf{x}_j$. This quantity can again be approximated by using the index-to-index relation $r(j)$ together with the categorical probabilities in P^k (c.f. Equation 4), such that $\phi_k(v_j; \mathbf{x}_j) \approx p^k_{r(j)}(\mathbf{x}_j)$. It is worthwhile to note that when working with discrete models, where the categories $\{f_1, \ldots, f_s\}$ have a one-to-one relation to the range of all simulation values in the training image, these approximations are exact. Finally, a computable ratio is provided by

$$\frac{\rho(\mathbf{m})}{\phi_k(\mathbf{m})} = \prod_{j=1}^{n_k} \frac{q_{r(j)}(\mathbf{x}_j)}{p^k_{r(j)}(\mathbf{x}_j)}. \tag{10}$$

This expression is very practical. All the quantities in Equation (10) are assembled during the PoPEx algorithm, so that the required effort for evaluating the ratio is negligible. Moreover, the expression is easily translated into log-probabilities, what can simplify the floating-point representation of the values. Although it only represents an approximation of the true ratio, often the assumptions are not too strongly violated, and the usage of the above equation is feasible. Finally, the weights w_k are computed by correcting the likelihood measure according to the hard conditioning data:

$$w_k = L(\mathbf{m}_k)\frac{\rho(\mathbf{m}_k)}{\phi_k(\mathbf{m}_k)} = L(\mathbf{m}_k)\prod_{j=1}^{n_k} \frac{q_{r(j)}(\mathbf{x}_j)}{p^k_{r(j)}(\mathbf{x}_j)}. \tag{11}$$

The ratios in the correction term compare the prior vs. the likelihood weighted probabilities of observing the selected values at the locations of the conditioning data. These quantities are directly available in the Q and P^k maps.

2.3.1. Degeneracy of the Sampling Weights

The estimator $\tilde{\mu}$ in Equation (8) suffers from a degeneracy in the sense that the distribution of $W^N = \{w_1, \ldots, w_N\}$ may become increasingly skewed when the dimension of $\mathcal{M}$ grows large (Doucet et al., 2001; Robert and Casella, 2004; Liu, 2008). This means, that the weights may take small values with high probability, but occasionally become very large. Using such weights in Equation (8) would produce estimators that are dominated by very few samples. Several preventive techniques exist, and they often try to consider a reduced dimensionality in the computation of the weights (Doucet et al., 2001; Rubinstein and Kroese, 2016). The expression in Equation (10) uses a reduction technique by limiting the computation of the ratio to the hard conditioning data. But this expression only represents one part of the weights in Equation (11), so that the degeneracy problem still exists. A diagnostic that can be used to assess the skewness of the weights, is called the **effective sample size** (Owen, 2013) and defined as

$$n_e(W^N) = \frac{\left(\sum_{i=1}^N w_i\right)^2}{\sum_{i=1}^N w_i^2} = \frac{N\overline{w}^2}{\overline{w^2}}, \tag{12}$$

where $\overline{w} = (1/N)\sum_{i=1}^N w_i$ and $\overline{w^2} = (1/N)\sum_{i=1}^N w_i^2$. There is an obvious link between n_e and the variance of W^N. It suffices to notice, that an estimator of the variance is obtained by $\overline{w^2} - \overline{w}^2$. Strongly varying weights would give $\overline{w}^2/\overline{w^2} << 1$ and therefore, $n_e << N$. In general, lowering the variance increases the effective number n_e. In practice, it is often hard to specify a bound under which n_e is alarmingly small, because this strongly depends on the application.

We will now present a method that aims to soften the degeneracy by modifying the variance of the weights. The value of any positive weight $w_i > 0$ can be changed by exponentiation, $(w_i)^\alpha$, and we know that

$$\lim_{\alpha\to 0}(w_i)^\alpha = 1.$$

For a given $0 < \alpha < 1$, the variance can therefore be reduced by transforming the set W^N into

$$W^N_\alpha = \{(w_1)^\alpha, \ldots, (w_N)^\alpha\}.$$

It is clear that in the limit $\alpha \to 0$, $n_e(W^N_\alpha)$ is equal to the total number of positive weights in W^N. Before computing an estimator $\hat{\mu}$ from W^N, we select an appropriate α, and use the weights in W^N_α instead. To make a good choice for α might depend on the application and can be challenging. We propose to define a lower bound l_0 and choose α such that

$$n_e(W^N_\alpha) = \max\left\{l_0,\, n_e(W^N)\right\}. \tag{13}$$

The idea of Equation (13) is to ensure that the computation of $\hat{\mu}$ is based on at least l_0 significant models. Furthermore, it assures that the growth rate of $n_e(W^N)$ and $n_e(W^N_\alpha)$ are equal for $n_e(W^N) > l_0$. This might be important for the asymptotic behavior of the method. Finally, we propose to use the pseudo code in the algorithm 3 to compute predictions. The computation of α can be translated into a smooth, 1-dimensional optimization problem, and does not require a considerable effort. The most important effort usually goes into the evaluation of $f(\mathbf{m}_i)$. But all the weights are known in advance and therefore we can omit computations that are associated with zero weights. Furthermore, the iterations in the algorithm 3 are independent and can be performed simultaneously in parallel.

Algorithm 3 Prediction

```
Input: $l_0$ and $f(\cdot)$
Output: $\hat{\mu}$
compute $\alpha$ such that $n_e(W^N_\alpha) = \max\{l_0, n_e(W^N)\}$
for $(w_i)^\alpha > 0$ do
  compute $f(\mathbf{m}_i)$
end for
compute $\hat{\mu}$
```

3. CASE STUDY AND RESULTS

In this section, we illustrate how PoPEx performs to solve an inverse problem with an example of a tracer test in a fluvial aquifer. We also consider the problem of quantifying the uncertainty related to the prediction of the capture zone of a pumping well in such geological environments.

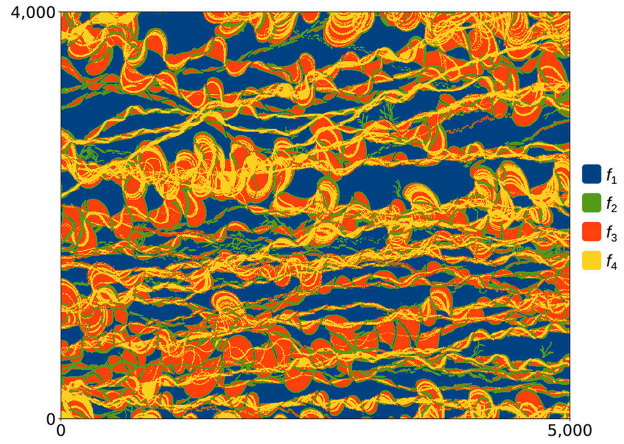


FIGURE 2 | Training image.

3.1. Problem Setup

For this example, the conceptual model for the geological heterogeneity is derived from a 3D simulation of the geological processes occurring in a fluvial plain using the FLUMY software (Lopez et al., 2009). This tool combines a process-based approach with a stochastic component. A meandering river crosses an alluvial valley with a given slope and causes erosion and deposition of sediments. Over time, the river migrates and alter the topography of the alluvial plain. This process generates complex geological patterns with a realistic and highly heterogeneous architecture. However, the thickness of the alluvial sediments is usually negligible with respect to the horizontal dimensions of the plain. It is therefore reasonable to reduce the complexity of the problem and neglect the vertical component of the flow. The parameter of interest is then the transmissivity of the aquifer.

Following this approach a 2-dimensional training image was generated. It is represented in **Figure 2**. The training image represents a domain of $5,000 \times 4,000\ m$ and is subdivided into $1,000 \times 800$ quadratic pixels. It was obtained by vertical integration of a 3D model generated with FLUMY. The resulting field was categorized into four facies types f_1,f_2,f_3, and f_4 that represent transmissivity values of $10^{-5}, 10^{-3}, 10^{-2}$, and $10^{-1}(m^2/s)$, respectively. The drainage porosity and the specific storage were fixed uniformly to 0.2 and 10^{-6}.

We then consider a smaller area of size $1,000 \times 500\ m$, discretized into 200×100 quadratic pixels. The area hosts a pumping well at the location $(750, 250)$ that extracts $15(l/s)$ of groundwater for a total duration of 20 days. The terrain is exposed to a natural slope of 4‰ in the x-direction, while the basin is closed at $y = 0\ (m)$ and $y = 500\ (m)$. Corresponding boundary conditions are: fixed head values of $4\ (m)$ $(left)$ and $0\ (m)$ (right) together with no-flow on the upper and lower boundary. A constant tracer concentration of $1(kg/m^3)$ is enforced at $(250, 250)$ throughout the time period.

For any given model, the subsurface water flow together with the tracer expansion is computed by the GroundWater simulation software (Arpat and Caers, 2007). At days 2, 4, 6, 8, 10, 12, 15, and 20 the solute concentration is recorded at the pumping well. This provides a set of 8 observations and represents all the data constraints used for conditioning the inverse problem.

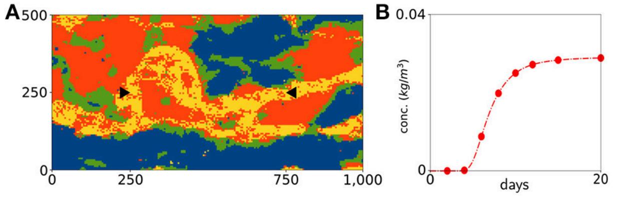


FIGURE 3 | (A) Shows the reference domain with tracer injection (left) and pumping well (right), while **(B)** is the observed tracer concentration at the pumping well.

For modeling the spatial structures, we used the training image shown in **Figure 2** and the DeeSse multiple-point statistics software (Straubhaar, 2011). An arbitrary seed was used to generate the reference domain in **Figure 3A**. The black triangles indicate tracer injection (left, pointing right) and pumping well location (right, pointing left), respectively. The tracer concentration at the pumping well resulting from this reference domain is shown in **Figure 3B**. The red dots indicate the extracted data that was used in the inverse procedure. This means that the entire reference domain in **Figure 3A** is unknown to the PoPEx algorithm. Its only task is to represent an unknown subsurface model and provide a sparse set of data points that can be used in the sampling procedure. For constructing the likelihood measure $L(\mathbf{m})$, we assume the observations to be independent and consider a multivariate normal distribution between the predictions $\mathbf{g}(\mathbf{m}) = \{g_1(\mathbf{m}), \ldots, g_8(\mathbf{m})\}$ and observations $\mathbf{d}^{\text{obs}} = \{d_1^{\text{obs}}, \ldots, d_8^{\text{obs}}\}$ with uniform standard deviation of $\sigma_L = 0.0015(kg/m^3)$. This represents 1.5‰ of the concentration at the injection point, and roughly 5% of the maximal concentration at the extraction location in the reference domain (c.f. **Figure 3B**). The subscript L distinguishes the standard deviation of the likelihood measure σ_L from the posterior density σ in Equation (1). Assuming an uniform and independent Gaussian behavior of $\mathbf{g}(\mathbf{m})$ around $\mathbf{d}^{\text{obs}}$, the density function of the likelihood measure is proportional to $\exp\left\{-\frac{1}{2\sigma_L^2}\sum_i (g_i(\mathbf{m}) - d_i^{\text{obs}})^2\right\}$.

3.2. Tracer Breakthrough Curve

The PoPEx method has been trained to run the above problem for a total of $N = 20,000$ models with $n_{\max} = 25$. Three random realizations are shown in **Figure 4**.

The prior facies probabilities in Q were computed from 500 unconditioned MPS models. For each realization in the PoPEx chain, the algorithm computed the tracer concentration at the pumping well, extracted 8 data points and compared them to the reference data in **Figure 3B**. Together with the weights from Equation (11), the posterior distribution of the tracer breakthrough curve can be computed. **Figure 5** shows the 2.5 –

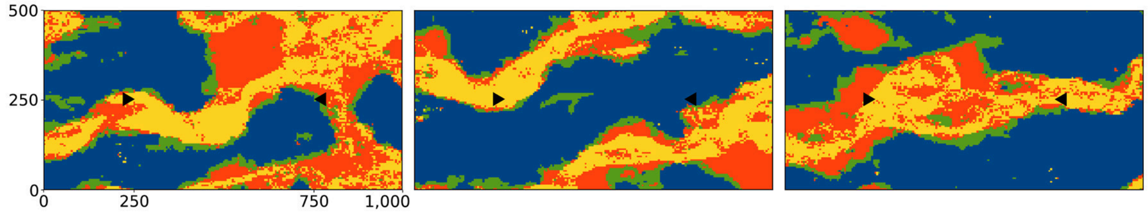


FIGURE 4 | Three random realizations from the PoPEx sampling. The above problem assigns a large likelihood value to the left, a very low likelihood value to the middle, and an average likelihood value to the right parameter map.

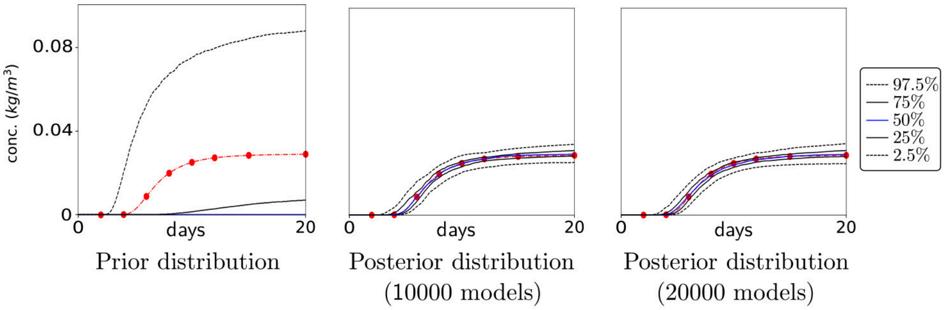


FIGURE 5 | Prior and posterior concentration probabilities at the pumping well. The curves indicate 2.5 – 97.5% region (dashed), 25 – 75% region (full) and average value (blue), while the red dots represent the reference concentration.

97.5% (dashed), 25 — 75% (full) and average (blue) curves of the prior and the posterior tracer concentration at the pumping well.

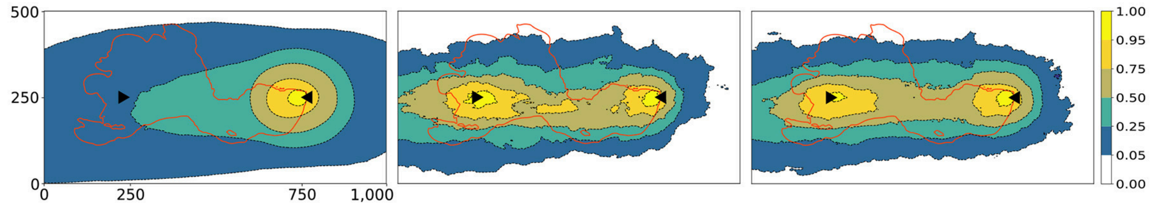
The red dots indicate the extracted reference data. It is clear that for any sampling strategy a critical measure is the required computational effort, which usually is proportional to the number of samples. For this reason, all results are shown for two different stages in the sampling procedure: after 10, 000 and after 20, 000 realizations. At a first glance, both estimations of the posterior probabilities are quite similar. This may be surprising when keeping in mind that the computational effort for the second estimation is twice as high. However, it can be seen that the probability lines are steadier and smoother in the last image. Both estimations of the 50% region (between the full lines) fully embeds the red reference data and follows the shape of the reference curve very precisely. The estimation of the posterior expectation (blue) almost matches the entire curve. The higher density of data points in the first 10 days, increases the relative importance of this period with respect to the second half. Thus, it is reasonable to allow less uncertainty in the beginning of the simulation. The more generous 95% regions (between the dashed curves) are still appropriate in reproducing the shape of the reference curve. This is even more significant when realizing that the prior distribution is far from being centered around the reference curve.

3.3. Predict 10-Days Capture Zone

In practice, when producing freshwater from an aquifer, it is often crucial to protect the resource and determine the capture zone (Leeuwen et al., 1998). Here, we used the results of the PoPEx model chain for predicting the posterior probabilities of the 10-days capture zone. It means that for each location in the simulation grid, we computed a Bernoulli probability value for the water to be captured within 10 days. **Figure 6** shows the predicted probabilities for the prior distribution and for the posterior distribution after 10, 000 and 20, 000 iterations, respectively.

As expected, since the tracer is arriving in <10 days at the pumping well, the injection point is located within a region having a high probability to belong to the 10-days capture zone. This is already clearly visible in the map generated from 10, 000 realizations. These results show the existence of a connected path of high transmissivity between the injection point and the pumping well. However, zones of lower probability are located in between these two points. This indicates that the position of the channel is not well identified from these tracer data alone. In the reference domain, shown in **Figure 3**, we can see that the yellow facies (with the largest transmissivity value) first shows a very tight upwards bend before heading almost directly toward the extraction well. The injected tracer will mostly follow the region with the largest transmissivity. Therefore, it will not take a direct path toward the well and its arrival time will be delayed. The only information that can be extracted from the observations is the delay. From the available data, it is therefore impossible to predict precisely water pathways that are far from the tracer injection and the algorithm is correctly informing us about that uncertainty.

It is interesting that the reference capture zone (red line) slightly passes outside the 95% region in the top section of the



Prior distribution

Posterior distribution (10000 models)

Posterior distribution (20000 models)

FIGURE 6 | Prior and posterior 10 days capture zone probabilities. The curves indicate 5, 25, 50, 75, and 95% regions, while the red lines delineates the capture zone in the reference domain.

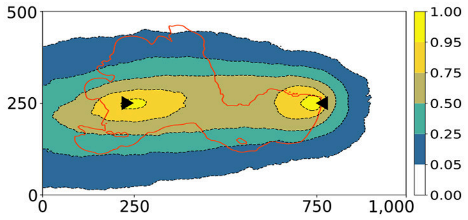


FIGURE 7 | Exact prediction of the 10-days capture zone probability. The red lines delineates the capture zone in the reference domain.

computational domain. This should not be interpreted as an inaccuracy of the PoPEx method, because it similarly appears in the approximation of the exact solution in **Figure 7**. However, it indicates that the training image (prior knowledge) together with the available observations (likelihood function) make the upwards extension of the reference zone very unlikely in terms of the posterior probability.

3.4. Convergence and Parallel Behavior

The synthetic inverse problem described above allows to compute exact predictions from a sufficiently large set of models. To do so, we put $n_{\max} = 0$ and generated an empirical reference set of $1,000,000$ unconditioned realizations. From this large ensemble, any prediction can be computed accurately by using Equation (8) together with weights such that $w_k = L(\mathbf{m}_k)$ (c.f. Equation 11). As the reference set is sufficiently large, the degeneracy problem described in section 2.3 can be ignored. The resulting predictions are considered to be the exact solutions and are denoted by μ_{ex}. Although the number of realizations is very large, it is not unsoiled to call these solutions to be exact. Nevertheless, these are very accurate approximations of the true solution such that, in this work, we will call them "exact prediction" or "exact solution." The corresponding prediction of the 10-days capture zone probability is shown in **Figure 7**.

Once an exact solution is available, we might be interested in the convergence speed of the PoPEx algorithm. Therefore, after each iteration $k = 1, \ldots, N$, a prediction $\hat{\mu}_k$ is computed by using the algorithm 3 and compared to μ_{ex}. As mentioned earlier, these two maps define Bernoulli probability values for each point in the computational domain. It determines whether the groundwater

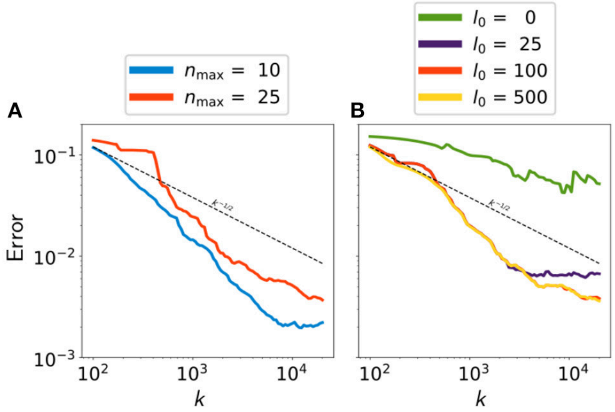


FIGURE 8 | Error between $\hat{\mu}_k$ and μ_{ex} for a fixed $l_0 = 100$ and variable $n_{\max}$ **(A)**, and fixed $n_{\max} = 25$ and variable l_0 **(B)**.

at the corresponding location belongs to the 10 days capture zone or not. A convenient distance between two Bernoulli probability maps $\hat{\mu}_k$ and μ_{ex} is the Jensen-Shannon divergence (JSD) [e.g., Lin (2006)] reading

$$J(\hat{\mu}_k || \mu_{\text{ex}}) = \frac{1}{2}\Big(D(\hat{\mu}_k || m) + D(\mu_{\text{ex}} || m)\Big),$$

with $m = (\hat{\mu}_k + \mu_{\text{ex}})/2$ and D being the Kullback-Leibler divergence as in Equation (5). This distance measure is computed pointwise over the simulation grid and therefore defines one distance value per pixel. A (scalar) error value is then obtained by computing the spatial average of the Jensen-Shannon divergence map.

Figure 8 shows the evolution of error between $\hat{\mu}_k$ and μ_{ex} with respect to the iteration k. For increasing the statistical significance of the results, every curve represents the average performance of 5 similar runs with different initial seed. First, the minimum number of effective models l_0 has been fixed to 100 and we varied the maximum number of conditioning values $n_{\max} \in \{10, 25\}$. **Figure 8A** shows that the two convergence curves are quite similar. This is not surprising, because the PoPEx algorithm is designed to correct the influence of the hard conditioning by using Equation (10). It follows that for a reasonable hard conditioning bound, the results are not highly sensible to the choice of $n_{\max}$. On the other hand, it can be seen from the blue curve that for $n_{\max} = 10$ and $k > 9,000$ the error reaches a "plateau." This signifies that for a certain time, the PoPEx algorithm was not able to further improve the prediction or in other words, that the method could not find sufficiently important realizations. From such behavior it can be deduced that the learning effect must be reinforced by increasing $n_{\mathbf{max}}$. However, what is important is that the overall convergence rate of both curves well compares with the dashed line representing $k^{-1/2}$. This is significant because if we directly sample from the posterior probability distribution σ and k is the number of samples, the Central Limit Theorem (CLT) (Durrett, 2010) predicts a convergence rate of $k^{-1/2}$. Because the error curves represent the average performance of 5 PoPEx runs, it is not surprising that they slightly fluctuate and do not reproduce the theoretical rate of $k^{-1/2}$ precisely.

For the second experience, we fixed $n_{\max} = 25$ and varied $l_0 \in \{0, 25, 100, 500\}$. We recall that the choice of a large value for l_0 generally increases the effective number of weights but implies a risk to produce biased predictions. On the other hand, when the effective number of weights is too low, the predictions will be based on very few models and may be biased as well. It is therefore not surprising that for $l_0 = 0$ the approximation accuracy is very bad (green curve in **Figure 8B**). However, the remaining three convergence curves are highly similar for $k \leq 4,000$ where the magenta curve ($l_0 = 25$) reaches a "plateau" and has difficulties to further improve the approximations. As in the previous figure, the curved represent the average performance of 5 similar PoPEx runs with different initial seed. It follows that small fluctuations may arise and should not be overestimated. However, the stagnation of the curve with $l_0 = 25$ might be due out of a different reason. Whenever the parameter l_0 is small, the weights in W_α^k are more sensible to highly dominant values. This means that a model $\mathbf{m}^{k_0}$ with very large weight w_{k_0} might dominate the prediction $\hat{\mu}^k$ for many iteration $k > k_0$ and therefore, the approximation error only slightly changes. So such a behavior indicates that l_0 should not be too small. We can again conclude by the fact for a reasonably large l_0, the overall convergence rate compares very well with the theoretical rate of $k^{-1/2}$.

The last part of the results section is dedicated to a short analysis of the parallel scalability of PoPEx. We repeat the same exercise by first using $n_{\text{par}} = 15$ on a 64 CPU facility ($34.4(Tflop/s)$), and then changing to $n_{\text{par}} = 75$ on 320 CPU's ($172(Tflop/s)$). Therefore, between the first and the second procedure, the computational capacity has been increased by a factor of 5. The performances will be compared by measuring the total sampling time and by a convergence analysis similar to the one in **Figure 8**. We fixed $n_{\max} = 25$ and $l_0 = 100$ and ran PoPEx until 20,000 models have been sampled. All runs were performed 5 times with different initial seeds. The total runtime for the two setups was $27.51 \pm 1.521[h]$ and $5.00 \pm 0.397[h]$, respectively. This signifies an overall speedup factor of 5.5 ± 0.74 and therefore fully satisfies the expectations.

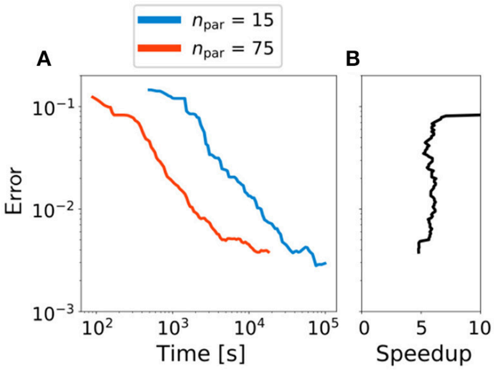


FIGURE 9 | Error between $\hat{\mu}_k$ and μ_{ex} for a fixed $n_{\max} = 25$, $l_0 = 100$ and variable n_{par} **(A)**, and the speedup factor for obtaining the same error accuracy **(B)**.

Considering the convergence analysis in **Figure 8** we are now interested in the speedup factor for obtaining the same approximation accuracy when predicting the 10-days capture zone probability. This means that in each iteration $k = 1, \ldots, 20,000$, the approximation errors are again computed by a Jensen-Shannon divergence between the prediction and the exact solution. In **Figure 9A** however, we compare the approximation error vs. the elapsed time in (s).

It can be seen that the convergence rate of both curves are very similar and the obtained gain factor highly matches the increasement of the computer resources. This becomes even more obvious in **Figure 9B**. It shows the observed speedup in time for obtaining the same approximation accuracy. This means that for any error value (y-axis) we computed the times (and the corresponding speedup factor) that were needed for reaching the considered approximation accuracy. From the relatively small statistical set of 5 chains per exercise, it is not surprising that there is a certain variability in the computations. However, it is evident that the curve significantly matches the predicted speedup factor of 5 and therefore underlines the exceptional scaling behavior of the PoPEx algorithm.

4. DISCUSSION

This paper presents a fast and efficient sampling method for solving inverse problems having a complex and discrete prior. The algorithm is parallelized and scales perfectly. This means that the number of samples computed in parallel is equal to the time reduction factor without compromising the quality of the results. Every sample involves two different main processes: generate a new model and compute the corresponding likelihood value. In this regard, the main concern for using the proposed method in practice is the number of such processes that can be run simultaneously. As there are many supercomputers publicly available however, handling a significant number of computations in parallel should not be a major issue.

Some important concepts of the above algorithm have originally been introduced by Jäggli et al. (2017) where the inverse method was named Posterior Population Expansion (PoPEx). In the present paper we suggest some minor changes concerning the sampling procedure and completely reconsider the method to compute predictions. Nevertheless, we decided to keep the name of the algorithm so that whenever the terminology *PoPEx* is used in the following, it refers to the algorithm as presented in this paper. PoPEx is capable to handle all the four different types of uncertainty distinguished by Sagar et al. (1975): spatial heterogeneities, initial conditions, boundary conditions and sources/sinks. The only requirement for the algorithm to be efficient, is that some uncertainties are modeled by conditional simulation tools.

As illustrated in the case study, a possibility is to use Multiple Point Statistics (MPS) to produce the conditional simulations of heterogeneity. But whenever MPS tools are used, a critical issue is to select an appropriate training image. In practice, it is therefore not uncommon to hesitate about this choice. With the above method, multiple training images can be included. This corresponds to a discrete choice that needs to be formulated in the inverse problem. PoPEx can iteratively learn which image is most appropriate and provide a posterior distribution of the training image selection issue.

PoPEx has been tested based on a two dimensional meandering channel aquifer of size $1,000 \times 500$ (m). A natural gradient of 4‰ and a groundwater extraction rate of $15(l/s)$ control the groundwater flow. Considering the high complexity of the categorical models, together with the small number of extracted data points, the method solved the inverse problem efficiently and produced accurate estimations of prediction uncertainty. After a very large computational effort, we were able to compute the exact solution and compare it with the predictions made by a PoPEx chain. It was shown empirically that the prediction converged to the exact solution very fast. The convergence speed was comparable with the theoretical rate of $k^{-1/2}$ predicted by the central limit theorem (where k is the number of samples). Furthermore, we demonstrated that the PoPEx results are not very sensitive to the choice of the two main input variables $n_{\max}$ and l_0. This is very convenient, because there is no uniform criterion for their optimal choice.

In section 2, we mentioned that PoPEx can be interpreted as an adaptive importance sampler (AIS). According to Oh and Berger (1992), the sampling distribution ϕ_k of an AIS technique should follow three properties:

- it should be easy to generate random samples from ϕ_k;
- the tails of ϕ_k should not be sharper than the tails of $f * \sigma$;
- ϕ_k should mimic $f * \sigma$ well.

The first property depends on the conditional simulation tool entrained to generate new models and is usually satisfied. Regarding the third property, it can be shown that the sampling distribution that minimizes the variance of $\hat{\mu}_k$ in Equation (8) is proportional to $f * \sigma$. When working with prior distributions ρ that are fairly flat over the region where $f(\mathbf{m})L(\mathbf{m})$ is concentrated, taking a sampling distribution proportional to $f * L$ is nearly optimal (Oh and Berger, 1992). But as samples may be used to generate predictions for many different functions, PoPEx is trying to learn a sampling distribution according to the likelihood values $L(\mathbf{m})$ (c.f. Equation 4). However, the link between L and ϕ_k must not be too strong. Let's assume that for a sufficiently large k, the sampling distributions is approximately proportional to L^r for a given power $r > 1$. In this case we have

$$\frac{\sigma}{\phi_k} \propto \frac{\rho}{L^{r-1}}.$$

For a flat distribution ρ and an infinite model space $\mathcal{M}$ this ratio might be unbounded so that the variance of Equation (8) is not finite.

The main limitation of the PoPEx method is that the likelihood values in Equation (8) must be evaluated and represented by a floating-point number. If the dimension of the data space is very large, it may happen that the numerical likelihood values are zero for most realizations. In this case, most of the indicator functions in Equation (4) are multiplied by zero and the learning process of the method is very slow. But if the number of observations is large, it is not uncommon that they are highly correlated. This means that it might be possible to trim the data set and project the observations onto a smaller data space. In other words, a possible strategy to overcome this issue would be to analyze the set of observations, extract a smaller amount of independent information and define an appropriate likelihood function. Alternatively, the likelihood function may be written as a *Gibbs field* (or *measure*) (Winkler, 2012), i.e.,

$$L(\mathbf{m}) = \frac{1}{C} \exp\{-H(\mathbf{m})\}.$$

Such a measure is induced by a normalization constant C and an energy function H. The latter is unique up to an additive constant and therefore, for finite model spaces as well as for Gaussian distributions we may assume that $H \geq 0$. Usually, for floating point operations it is easier to work with the energy $H(\mathbf{m})$ rather than with the unnormalized Gibbs measure $\exp\{-H(\mathbf{m})\}$ directly. During the evolution of the PoPEx algorithm, whenever P^k is computed from Equation (4), we could weight the indicator functions $\mathbf{1}_{f_i}(\mathbf{m}_j)$ proportional to

$$\frac{1}{1 + H(\mathbf{m}_j)}$$

rather than $\tilde{L}(\mathbf{m}_j)$. For the computation of ergodic predictions however, we would still need to compute the likelihood values. But considering the underlying floating point operations, it can be advantageous to learn from non-zero energy values $H(\mathbf{m})$ in order to obtain a sufficiently large number of non-zero likelihood values $L(\mathbf{m})$.

DATA AVAILABILITY

The raw data supporting the conclusions of this manuscript will be made available by the authors, without undue reservation, to any qualified researcher.

AUTHOR CONTRIBUTIONS

All authors conceived of the presented idea and developed the theory. CJ designed and wrote the software in order to perform the computations. All authors designed the case study, discussed the results and contributed to the final manuscript.

ACKNOWLEDGMENTS

We would like to thank Przemyslaw Juda for his contributions toward the generation of the conceptual training image. This work was founded by the Swiss National Science Foundation through the grant number 153637.

REFERENCES

Arpat, G. B., and Caers, J. (2007). Conditional simulation with patterns. *Math. Geol.* 39, 177–203. doi: 10.1007/s11004-006-9075-3

Carrera, J., Alcolea, A., Medina, A., Hidalgo, J., and Slooten, L. (2005). Inverse problem in hydrogeology. *Hydrogeol. J.* 13, 206–222. doi: 10.1007/s10040-004-0404-7

Chilès, J.-P., and Delfiner, P. (2009). *Geostatistics: Modeling Spatial Uncertainty, Vol. 497*. Hoboken, NJ: John Wiley & Sons.

de Marsily, G., Delhomme, J. P., Coudrain-Ribstein, A., and Lavenue, A. M. (2000). Four decades of inverse problems in hydrogeology. *Geol. Soc. Am. Spec. Pap.* 348, 1–17. doi: 10.1130/0-8137-2348-5.1

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice.* New York, NY: Springer.

Durrett, R. (2010). *Probability: Theory and Examples*. *Cambridge Series in Statistical and Probabilistic Mathematics.* New York, NY: Cambridge University Press.

Feyen, L., and Caers, J. (2006). Quantifying geological uncertainty for flow and transport modeling in multi-modal heterogeneous formations. *Adv. Water Resour.* 29, 912–929. doi: 10.1016/j.advwatres.2005.08.002

Fu, J., and Gómez-Hernández, J. (2008). Preserving spatial structure for inverse stochastic simulation using blocking markov chain monte carlo method. *Inverse Probl. Sci. Eng.* 16, 865–884. doi: 10.1080/17415970802015781

Gómez-Hernández, J. J., and Wen, X.-H. (1998). To be or not to be multi-Gaussian? a reflection on stochastic hydrogeology. *Adv. Water Resour.* 21, 47–61. doi: 10.1016/S0309-1708(96)00031-0

Hansen, T. M., Cordua, K. S., and Mosegaard, K. (2012). Inverse problems with non-trivial priors: efficient solution through sequential Gibbs sampling. *Comput. Geosci.* 16, 593–611. doi: 10.1007/s10596-011-9271-1

Hesterberg, T. C. (2003). *Advances in Importance Sampling*. Ph.D. thesis, Stanford University.

Honarkhah, M., and Caers, J. (2010). Stochastic simulation of patterns using distance-based pattern modeling. *Math. Geosci.* 42, 487–517. doi: 10.1007/s11004-010-9276-7

Jäggli, C., Straubhaar, J., and Renard, P. (2017). Posterior population expansion for solving inverse problems. *Water Resour. Res.* 53, 2902–2916. doi: 10.1002/2016WR019550

Journel, A., and Zhang, T. (2006). The necessity of a multiple-point prior model. *Math. Geol.* 38, 591–610. doi: 10.1007/s11004-006-9031-2

Kerrou, J., Renard, P., Franssen, H.-J. H., and Lunati, I. (2008). Issues in characterizing heterogeneity and connectivity in non-multigaussian media. *Adv. Water Resour.* 31, 147–159. doi: 10.1016/j.advwatres.2007.07.002

Kullback, S., and Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Stat.* 22, 79–86. doi: 10.1214/aoms/1177729694

Laloy, E., Linde, N., Jacques, D., and Mariethoz, G. (2016). Merging parallel tempering with sequential geostatistical resampling for improved posterior exploration of high-dimensional subsurface categorical fields. *Adv. Water Resour.* 90, 57–69. doi: 10.1016/j.advwatres.2016.02.008

Leeuwen, M., te Stroet, C. B. M., Butler, A. P., and Tompkins, J. A. (1998). Stochastic determination of well capture zones. *Water Resour. Res.* 34, 2215–2223. doi: 10.1029/98WR01552

Lin, J. (2006). Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.* 37, 145–151. doi: 10.1109/18.61115

Linde, N., Renard, P., Mukerji, T., and Caers, J. (2015). Geological realism in hydrogeological and geophysical inverse modeling: a review. *Adv. Water Resour.* 86, 86–101. doi: 10.1016/j.advwatres.2015.09.019

Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing.* New York, NY: Springer Publishing Company, Incorporated.

Lopez, S., Cojan, I., Rivoirard, J., and Galli, A. (2009). *Process Based Stochastic Modelling: Meandering Channelized Reservoirs.* Hoboken, NJ: Wiley-Blackwell, 139–144.

Mariethoz, G., and Caers, J. (2014). *Front Matter.* Hoboken, NJ: Wiley-Blackwell.

Mariethoz, G., Renard, P., and Caers, J. (2010a). Bayesian inverse problem and optimization with iterative spatial resampling. *Water Resour. Res.* 46:W11530. doi: 10.1029/2010WR009274

Mariethoz, G., Renard, P., and Straubhaar, J. (2010b). The direct sampling method to perform multiple-point geostatistical simulations. *Water Resour. Res.* 46:W11536. doi: 10.1029/2008WR007621

Moles, C. G., Mendes, P., and Banga, J. R. (2003). Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res.* 13, 2467–2474. doi: 10.1101/gr.1262503

Mosegaard, K., and Tarantola, A. (2002). "16 - probabilistic approach to inverse problems," in *International Handbook of Earthquake and Engineering Seismology, Vol. 81, Part A of International Geophysics,*, eds P. C. J. William, H. K. Lee, H. Kanamori, and C. Kisslinger (Philadelphia, PA: Academic Press), 237–265.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press.

Naylor, J., and Smith, A. (1988). Econometric illustrations of novel numerical integration strategies for Bayesian inference. *J. Econom.* 38, 103–125. doi: 10.1016/0304-4076(88)90029-2

Oh, M.-S., and Berger, J. O. (1992). Adaptive importance sampling in monte carlo integration. *J. Stat. Comput. Simul.* 41, 143–168. doi: 10.1080/00949659208810398

Oliver, D. S., Cunha, L. B., and Reynolds, A. C. (1997). Markov chain monte carlo methods for conditioning a permeability field to pressure data. *Math. Geol.* 29, 61–91. doi: 10.1007/BF02769620

Owen, A. B. (2013). *Monte Carlo Theory, Methods and Examples.* Available online at: https://statweb.stanford.edu/~owen/mc/

Robert, C. P., and Casella, G. (2004). "Monte carlo statistical methods," in *Springer Texts in Statistics* (New York, NY: Springer-Verlag).

Romary, T. (2010). History matching of approximated lithofacies models under uncertainty. *Comput. Geosci.* 14, 343–355. doi: 10.1007/s10596-009-9166-6

Rubinstein, R. Y., and Kroese, D. P. (2016). *Simulation and the Monte Carlo Method, 3 Edn*. Hoboken, NJ: Wiley.

Sagar, B., Yakowitz, S., and Duckstein, L. (1975). A direct method for the identification of the parameters of dynamic nonhomogeneous aquifers. *Water Resour. Res.* 11, 563–570. doi: 10.1029/WR011i004p00563

Straubhaar, J. (2011). *DeeSse Technical Reference Guide*. Technical Report, Centre d'hydrogéologie et géothermie, University of Neuchâtel, (Neuchâtel).

Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R., and Besson, O. (2011). An improved parallel multiple-point algorithm using a list approach. *Math. Geosci.* 43, 305–328. doi: 10.1007/s11004-011-9328-7

Straubhaar, J., Walgenwitz, A., and Renard, P. (2013). Parallel multiple-point Statistics algorithm based on list and tree structures. *Math. Geosci.* 45, 131–147. doi: 10.1007/s11004-012-9437-y

Strebelle, S. (2002). Conditional simulation of complex geological structures using multiple-point statistics. *Math. Geol.* 34, 1–21. doi: 10.1023/A:1014009426274

Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation* (Philadelphia, PA: Society for Industrial and Applied Mathematics)

Wainwright, J., and Mulligan, M. (2005). *Environmental Modelling: Finding Simplicity in Complexity*. West Sussex: John Wiley & Sons.

Winkler, G. (2012). *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*. *Stochastic Modelling and Applied Probability*. Berlin, Heidelberg: Springer.

Zhou, H., Gómez-Hernández, J. J., and Li, L. (2014). Inverse methods in hydrogeology: evolution and recent trends. *Adv. Water Resour.* 63, 22–37. doi: 10.1016/j.advwatres.2013.10.014

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.