

Water Resources Research

RESEARCH ARTICLE

10.1029/2018WR022658

Key Points:

- We propose an adaptive multifidelity MCMC algorithm
- The correlation relationship between the high- and low-fidelity models is utilized
- Most of the computational budget is distributed in the posterior region

Correspondence to:

L. Zeng,
lingzao@zju.edu.cn

Citation:

Zhang, J., Man, J., Lin, G., Wu, L., & Zeng, L. (2018). Inverse modeling of hydrologic systems with adaptive multifidelity Markov chain Monte Carlo simulations. *Water Resources Research*, 54, 4867–4886. <https://doi.org/10.1029/2018WR022658>

Received 27 JAN 2018

Accepted 12 JUN 2018

Accepted article online 20 JUN 2018

Published online 21 JUL 2018

Inverse Modeling of Hydrologic Systems with Adaptive Multifidelity Markov Chain Monte Carlo Simulations

Jiangjiang Zhang¹ , Jun Man¹, Guang Lin² , Laosheng Wu³ , and Lingzao Zeng¹ 

¹Zhejiang Provincial Key Laboratory of Agricultural Resources and Environment, Institute of Soil and Water Resources and Environmental Science, College of Environmental and Resource Sciences, Zhejiang University, Hangzhou, China,

²Department of Mathematics and School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA,

³Department of Environmental Sciences, University of California, Riverside, CA, USA

Abstract Markov chain Monte Carlo (MCMC) simulation methods are widely used to assess parametric uncertainties of hydrologic models conditioned on measurements of observable state variables. However, when the model is CPU-intensive and high dimensional, the computational cost of MCMC simulation will be prohibitive. In this situation, a CPU-efficient while less accurate low-fidelity model (e.g., a numerical model with a coarser discretization or a data-driven surrogate) is usually adopted. Nowadays, multifidelity simulation methods that can take advantage of both the efficiency of the low-fidelity model and the accuracy of the high-fidelity model are gaining popularity. In the MCMC simulation, as the posterior distribution of the unknown model parameters is the region of interest, it is wise to distribute most of the computational budget (i.e., the high-fidelity model evaluations) therein. Based on this idea, in this paper we propose an adaptive multifidelity MCMC algorithm for efficient inverse modeling of hydrologic systems. In this method, we evaluate the high-fidelity model mainly in the posterior region through iteratively running MCMC based on a Gaussian process system that is adaptively constructed with multifidelity simulation. The error of the Gaussian process system is rigorously considered in the MCMC simulation and gradually reduced to a negligible level in the posterior region. Thus, the proposed method can obtain an accurate estimate of the posterior distribution with a small number of the high-fidelity model evaluations. The performance of the proposed method is demonstrated by three numerical case studies in inverse modeling of hydrologic systems.

1. Introduction

For a better understanding and management of hydrologic systems, there is a growing interest in applying numerical modeling techniques to conduct qualitative and quantitative analyses (Anderson et al., 2015; Vieux, 2001). However, the existence of uncertainties in model structure, model parameters, initial and boundary conditions, measurement data, etc., would hinder the predictive accuracy of hydrologic modeling (Clark et al., 2011; Refsgaard et al., 2012; Wagener & Gupta, 2005). To reduce the predictive uncertainty of the hydrologic system of concern, it is common practice to calibrate the conceptual model against measurements of some state variables, for example, hydraulic head, solute concentration, temperature, and streamflow, through solving an inverse problem (Hu et al., 2017; Kang et al., 2017; Zha et al., 2018; Zhu et al., 2017).

When handling uncertainties in hydrologic modeling, inverse methods based on Bayes' theorem are appropriate options as they can be formulated in a coherent and consistent manner (Stuart, 2010; Vrugt, 2016). It means that each time we run the Bayesian method, it should converge to the same distribution. In the Bayesian framework, quantities of interest are modeled as random variables, whose posterior distribution is proportional to the prior distribution times the likelihood. In most situations, closed-form expressions of the posterior distribution are nonexistent, so one has to resort to Monte Carlo simulation methods to obtain numerical approximations. Over the past decades, Markov chain Monte Carlo (MCMC) methods have been widely used to assess uncertainties of hydrologic systems conditioned on measurements of observable state variables (Shi et al., 2014; T. J. Smith & Marshall, 2008; Vrugt, 2016; L. Zeng et al., 2012; X. Zeng et al., 2018; Zhang et al., 2015; Zhang et al., 2016). However, MCMC has to sufficiently explore the parameter space to obtain reliable estimation results. This usually requires a large number of model evaluations. However, it is common that the execution time of a single forward model simulation can be in hours or even longer, for example, for distributed groundwater models (Elshall & Tsai, 2014; Keating et al., 2010). In this situation, the computational cost of MCMC simulation will be prohibitive, especially for high-dimensional problems.

To alleviate the computational cost, one can use a CPU-efficient low-fidelity model (denoted by $f_L(\mathbf{m})$, where \mathbf{m} signifies the model parameters) of the original model (denoted by $f_H(\mathbf{m})$, that is, the high-fidelity model in this paper) in the MCMC simulation. A low-fidelity model could be a data-driven surrogate based on interpolation or regression, a numerical model that considers fewer processes or has a lower numerical precision (e.g., with a coarser discretization) or is constructed by projecting high-dimensional variables onto their low-dimensional subspace, etc. (Asher et al., 2015; Mo et al., 2017; Razavi et al., 2012; R. C. Smith, 2014; X. Zeng et al., 2018; Zhang et al., 2017). Inevitably, using the low-fidelity model $f_L(\mathbf{m})$ can introduce some bias if no error model is considered (Forrester & Keane, 2009; Razavi et al., 2012). To address this issue, one popular approach is to further evaluate $f_H(\mathbf{m})$ in a two-stage manner; that is, first explore the parameter space sufficiently using $f_L(\mathbf{m})$ with a low computational cost at stage one, then use $f_H(\mathbf{m})$ to correctly sample the target distribution at stage two (Efendiev et al., 2005; Laloy et al., 2013; L. Zeng et al., 2012; Zhang et al., 2015). In this way, many unnecessary evaluations of $f_H(\mathbf{m})$ can be avoided. More recently, another approach that adaptively refines a data-driven surrogate over the posterior distribution is proposed, which has shown to be highly efficient (Gong & Duan, 2017; Ju et al., 2018; Zhang et al., 2016).

In the above approaches, nevertheless, the correlation relationship between $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ is not utilized, which leaves some potential untouched. One method that is suitable for fusing $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ into an integrated system is the multifidelity simulation (Kennedy & O'hagan, 2000). Given a small number of $f_H(\mathbf{m})$ evaluations and a much larger number of $f_L(\mathbf{m})$ evaluations, the multifidelity simulation can take advantage of both the efficiency of $f_L(\mathbf{m})$ and the accuracy of $f_H(\mathbf{m})$. The integrated system can be constructed with many methods, for example, polynomial chaos expansion (Narayan et al., 2014; Palar et al., 2016; Zhu et al., 2014) and Gaussian process (GP) (Kennedy & O'Hagan, 2000; Le Gratiet & Garnier, 2014; Parussini et al., 2017; Raissi et al., 2017). GP is a generic supervised learning method that uses a (multivariate) Gaussian distribution to predict the quantity of interest based on a set of training data (Williams & Rasmussen, 2006). GP has been widely used in hydrologic science to simulate the input–output relationship of the system model (Asher et al., 2015; Sun et al., 2014; Zhang et al., 2016) and the model structural error (Xu et al., 2017; Xu & Valocchi, 2015), etc. As GP is very flexible (i.e., we can specify different covariance functions) and can provide the uncertainty estimation (variance) of the system output, we adopt it in this paper to simulate the input–output relationship of $f_H(\mathbf{m})$ with multifidelity simulation. In the multifidelity GP framework, the correlation between $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ is rigorously considered with the covariance (kernel) functions, and the corresponding hyperparameters are estimated with an optimization method conditioned on simulation data of both $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$.

To improve the performance of the multifidelity GP system, people would like to acquire the training data adaptively via active learning. One intuitive strategy is to add a new set of training data that have the largest output variance (Raissi et al., 2017). Other strategies can be more sophisticated, for example, the expected informativeness of candidate points (Mackay, 1992), the mutual information criterion (Krause et al., 2008), and the integrated posterior variance (Gorodetsky & Marzouk, 2016), just name a few. How to acquire the optimal new training data adaptively for a specific problem is still an open problem. In many cases, the posterior occupies a very small proportion of the prior region. If the multifidelity system is built over the whole prior distribution, its accuracy cannot be guaranteed if only a limited number of $f_H(\mathbf{m})$ evaluations are affordable. Considering that the posterior distribution is the region of interest, it is wise to distribute most of the computational budget (i.e., $f_H(\mathbf{m})$ evaluations) therein. Based on this idea, we propose an adaptive multifidelity MCMC (AMF-MCMC) algorithm for efficient inverse modeling of hydrologic systems in this paper. Here we iteratively run MCMC with a GP system constructed with multifidelity simulation to search the posterior region and accordingly add new $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ data to refine the GP system locally. Gradually, the GP system will be accurate enough in the posterior region. Finally, we can obtain an accurate estimate of the posterior distribution. In the AMF-MCMC algorithm, most of the $f_H(\mathbf{m})$ simulations are run near or within the posterior region, thus very little computational budget is wasted.

It is noted that, multi-fidelity simulation methods, that is, using paired simple and complex models to achieve a balance between accuracy and efficiency, have become increasingly popular in hydrologic science (Doherty & Christensen, 2011; Linde et al., 2017; Lu et al., 2016; Mosleh et al., 2015; Watson et al., 2013), while the application of these methods in Bayesian inference is very limited. Here we adopt the method originally developed by Kennedy and O'Hagan (2000) to build the multifidelity GP system. The novelty of this paper is that we propose an efficient framework to build the multifidelity GP system adaptively in the posterior

distribution, not the traditional way in the prior distribution. The method proposed in this work can accurately sample the posterior distribution, but only requires a small number of $f_H(\mathbf{m})$ evaluations. To our best knowledge, this framework (i.e., refine the multifidelity surrogate adaptively over the posterior distribution) is rather new and it has value in both theoretical and practical aspects.

The remainder of this paper is organized as follows. In section 2, we provide a detailed formulation of the AMF-MCMC algorithm. Then, its performance is demonstrated by three numerical case studies in section 3. Finally, some conclusions and discussions are provided in section 4.

2. Methods

2.1. Bayesian Inference With DREAM_(zs)

For simplicity, here we represent the observation process of the hydrologic system of concern in the following compact form:

$$\tilde{\mathbf{d}} = f(\mathbf{m}) + \varepsilon, \quad (1)$$

where $\tilde{\mathbf{d}}$ is a N_d -vector for the measurements, $f(\cdot)$ is the system model, \mathbf{m} is a N_m -vector for the unknown model parameters, and ε is a N_d -vector for the measurement errors. Before obtaining the measurements $\tilde{\mathbf{d}}$, our knowledge about the unknown model parameters \mathbf{m} is represented by the prior distribution, $p(\mathbf{m})$. When $\tilde{\mathbf{d}}$ is available, we can update our knowledge about \mathbf{m} with the posterior distribution, $p(\mathbf{m}|\tilde{\mathbf{d}})$, according to Bayes' theorem:

$$p(\mathbf{m}|\tilde{\mathbf{d}}) = \frac{p(\mathbf{m})p(\tilde{\mathbf{d}}|\mathbf{m})}{p(\tilde{\mathbf{d}})}, \quad (2)$$

where $L(\mathbf{m}|\tilde{\mathbf{d}}) = p(\tilde{\mathbf{d}}|\mathbf{m})$ is the likelihood function that quantifies the mismatch between the model outputs $f(\mathbf{m})$ and the measurements $\tilde{\mathbf{d}}$, $p(\tilde{\mathbf{d}}) = \int p(\tilde{\mathbf{d}}|\mathbf{m})p(\mathbf{m})d\mathbf{m}$ is the evidence. When the measurement errors are assumed to fit multivariate Gaussian distribution, the likelihood $L(\mathbf{m}|\tilde{\mathbf{d}})$ can be expressed as

$$L(\mathbf{m}|\tilde{\mathbf{d}}) = \frac{1}{(2\pi)^{N_d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2} [\tilde{\mathbf{d}} - f(\mathbf{m})]^\top \Sigma^{-1} [\tilde{\mathbf{d}} - f(\mathbf{m})]\right\}, \quad (3)$$

where Σ is the covariance of the measurement errors, $|\cdot|$ signifies the determinant operator.

In most situations, analytical forms of $p(\mathbf{m}|\tilde{\mathbf{d}})$ do not exist. Here we resort to an efficient MCMC algorithm, that is, DREAM_(zs) (Laloy et al., 2013; Laloy & Vrugt, 2012; Vrugt, 2016), to explore the parameter space and estimate $p(\mathbf{m}|\tilde{\mathbf{d}})$ numerically. MCMC works by constructing a Markov chain that gradually converges to the posterior distribution. To better explore the parameter space, N_c parallel chains are generated and evolved simultaneously by DREAM_(zs). At iteration t , based on an archive of thinned chain history, \mathbf{Z} , the parallel direction jump and the snooker jump are used to update the previous state in the i th chain (i.e., \mathbf{m}_{t-1}^i) to obtain the proposal state, \mathbf{m}_p^i . Then, we will compare the acceptance rate, $p_{\text{acc}} = \min\left[1, p(\mathbf{m}_p^i|\tilde{\mathbf{d}})/p(\mathbf{m}_{t-1}^i|\tilde{\mathbf{d}})\right]$, with a random sample u drawn from the uniform distribution, $\mathcal{U}(0, 1)$. If $p_{\text{acc}} > u$, we will accept \mathbf{m}_p^i and let $\mathbf{m}_t^i = \mathbf{m}_p^i$; Otherwise, we will reject \mathbf{m}_p^i and let $\mathbf{m}_t^i = \mathbf{m}_{t-1}^i$. In every T_{thin} iterations, we will append the current N_c states in the Markov chains to the archive \mathbf{Z} . When the Markov chains converge to the stationary regime, we can view states in the chains as random samples drawn from the posterior distribution. For more details about DREAM_(zs) and related algorithms, one can refer to (Vrugt, 2016).

2.2. Multifidelity Simulation With Gaussian Process

To obtain reliable estimations, MCMC needs to sufficiently explore the parameter space, which generally requires a large number of model evaluations. When $f(\mathbf{m})$ is CPU-intensive (then $f(\mathbf{m})$ is called the high-fidelity model and represented by $f_H(\mathbf{m})$ thereafter), the computational cost of MCMC simulation will be prohibitive. In this situation, a CPU-efficient low-fidelity model $f_L(\mathbf{m})$ is usually adopted. To balance accuracy and

efficiency, it is desirable to construct an integrated system for the MCMC simulation through fusing the information provided by a small number of $f_H(\mathbf{m})$ evaluations and a much larger number of $f_L(\mathbf{m})$ evaluations. This can be realized with the autoregressive model (Kennedy & O'Hagan, 2000):

$$u_H(\mathbf{m}) = \rho u_L(\mathbf{m}) + \delta(\mathbf{m}), \quad (4)$$

where $u_L(\mathbf{m}) \sim \mathcal{GP}(0, k_1(\mathbf{m}, \mathbf{m}'; \phi_1))$ and $\delta(\mathbf{m}) \sim \mathcal{GP}(0, k_2(\mathbf{m}, \mathbf{m}'; \phi_2))$ are two independent GPs; $k_i(\mathbf{m}, \mathbf{m}'; \phi_i)$ are the covariance functions with hyperparameters ϕ_i , for $i = 1, 2$; ρ is the cross-correlation coefficient; \mathbf{m} and \mathbf{m}' are two arbitrary samples in the parameter space. Then $u_H(\mathbf{m})$ has the following form (Raissi et al., 2017):

$$u_H(\mathbf{m}) \sim \mathcal{GP}(0, k(\mathbf{m}, \mathbf{m}'; \phi)), \quad (5)$$

where $k(\mathbf{m}, \mathbf{m}'; \phi) = \rho^2 k_1(\mathbf{m}, \mathbf{m}'; \phi_1) + k_2(\mathbf{m}, \mathbf{m}'; \phi_2)$, and $\phi = [\phi_1, \phi_2, \rho]$. Consequently, we have

$$\begin{bmatrix} u_L(\mathbf{m}) \\ u_H(\mathbf{m}) \end{bmatrix} \sim \mathcal{GP}\left(0, \begin{bmatrix} k_{LL} & k_{LH} \\ k_{HL} & k_{HH} \end{bmatrix}\right), \quad (6)$$

where $k_{LL} = k_1(\mathbf{m}, \mathbf{m}'; \phi_1)$, $k_{LH} = k_{HL}^T = \rho k_1(\mathbf{m}, \mathbf{m}'; \phi_1)$, and $k_{HH} = k(\mathbf{m}, \mathbf{m}'; \phi)$. In this paper, we adopt the commonly used squared exponential covariance function (Williams & Rasmussen, 2006) for both k_1 and k_2 :

$$k_i(\mathbf{m}, \mathbf{m}'; \phi_i) = \sigma_i^2 \exp\left[-\frac{1}{2} \sum_{n=1}^{N_m} \frac{(\mathbf{m}_n - \mathbf{m}'_n)^2}{l_{n,i}^2}\right], \quad (7)$$

where $\phi_i = \left[\sigma_i^2, \left(l_{n,i}^2\right)_{n=1}^{N_m}\right]$ are the hyperparameters of the covariance functions k_i , for $i = 1, 2$.

When we have N_L sets of training data obtained by simulating $f_L(\mathbf{m})$ at N_L parameter samples and N_H sets of training data obtained by simulating $f_H(\mathbf{m})$ at N_H parameter samples (here N_L and N_H are two positive integers), we can estimate the hyperparameters ϕ by minimizing the negative log marginal likelihood:

$$NL = -\log p(\mathbf{D} | \mathbf{M}, \phi) = \frac{1}{2} \mathbf{D}^T \mathbf{K}^{-1} \mathbf{D} + \frac{1}{2} \log |\mathbf{K}| + \frac{N_L + N_H}{2} \log(2\pi), \quad (8)$$

where

$$\mathbf{K} = \begin{bmatrix} k_{LL}(\mathbf{M}_L, \mathbf{M}_L) + \sigma_L^2 \mathbf{I}_{N_L} & k_{LH}(\mathbf{M}_L, \mathbf{M}_H) \\ k_{HL}(\mathbf{M}_H, \mathbf{M}_L) & k_{HH}(\mathbf{M}_H, \mathbf{M}_H) + \sigma_H^2 \mathbf{I}_{N_H} \end{bmatrix}; \quad (9)$$

$\mathbf{M}_L = [\mathbf{m}_1, \dots, \mathbf{m}_{N_L}]$ are N_L parameter samples for $f_L(\mathbf{m})$, $\mathbf{D}_L = [f_L(\mathbf{m}_1), \dots, f_L(\mathbf{m}_{N_L})]$ are the corresponding low-fidelity model outputs; $\mathbf{M}_H = [\mathbf{m}_1, \dots, \mathbf{m}_{N_H}]$ are N_H parameter samples for $f_H(\mathbf{m})$, $\mathbf{D}_H = [f_H(\mathbf{m}_1), \dots, f_H(\mathbf{m}_{N_H})]$ are the corresponding high-fidelity model outputs; $\mathbf{M} = [\mathbf{M}_L \ \mathbf{M}_H]$, $\mathbf{D} = [\mathbf{D}_L \ \mathbf{D}_H]$; \mathbf{I}_{N_L} and \mathbf{I}_{N_H} are identity matrices of size N_L and N_H , respectively. Here the hyperparameters to be estimated are $[\phi, \sigma_L, \sigma_H]$. The above equation (8) calculates the goodness of fit given the training data and a set of hyperparameters, and it can be easily derived from the (logarithmic) probability density function of multivariate Gaussian distribution (it is noted here that the most general zero-mean function of GP is used; Williams & Rasmussen, 2006).

Minimizing equation (8) defines a nonconvex optimization problem. In practice, we use the trust-region algorithm, which is based on the interior reflective Newton method described in Coleman and Li (1996), to find the minimum solution. Although it does not guarantee convergence to a global optimum, it is usually sufficient for obtaining a good solution. The most computationally intensive part of solving the optimization problem is associated with inverting the covariance matrix \mathbf{K} , which scales cubically with the number of training data. This is a well-known limitation of GP, but it has been effectively addressed with techniques like sparse GP (Quinonero-Candela & Rasmussen, 2005; Snelson & Ghahramani, 2006). As will be demonstrated in section 2.3, the method proposed in this paper does not require a large number of training data. Moreover, the process of training GPs can be accelerated by adopting parallel computation. Thus, the computational cost of training GPs will not be a big problem.

After training the GP system conditioned on the multifidelity data \mathbf{D} [i.e., after obtaining the optimal hyperparameters by minimizing the objective function defined in equation (8)], we can obtain a new GP system represented by $\tilde{g}(\mathbf{m})$, which can be used to predict the model output given an arbitrary parameter set \mathbf{m}^* :

$$\tilde{g}(\mathbf{m}^*) = u_H(\mathbf{m}^*) \mid \mathbf{D} \sim \mathcal{N}\left(\mu_{gp}, \sigma_{gp}^2\right), \quad (10)$$

where $\mu_{gp} = \mathbf{aK}^{-1}\mathbf{D}$ is the mean estimate, $\sigma_{gp}^2 = k_{HH}(\mathbf{m}^*, \mathbf{m}^*) - \mathbf{aK}^{-1}\mathbf{a}^T$ is the estimation variance, and $\mathbf{a} = [k_{HL}(\mathbf{m}^*, \mathbf{M}_L) \ k_{HH}(\mathbf{m}^*, \mathbf{M}_H)]$. Then the multifidelity system $\tilde{g}(\mathbf{m})$ can be used in MCMC simulations to gain efficiency.

As the multifidelity GP (MF-GP) can fuse the efficiency of $f_L(\mathbf{m})$ and the accuracy of $f_H(\mathbf{m})$, it is advantageous over the GP system constructed only on data from $f_H(\mathbf{m})$ (here we call it the single-fidelity GP, i.e., SF-GP). As shown in Figure 1, in a one-dimensional problem, with three training data from the high-fidelity model and 20 training data from the low-fidelity model, we can build an MF-GP system that is more accurate than the SF-GP system based on four training data from the high-fidelity model, especially in the area that is far away from the high-fidelity training data. Here the high-fidelity model is $f_H(m) = \sin(m)$, and the low-fidelity model is obtained by adding an error term, $-0.1m - 0.1$, to the high-fidelity model, that is, $f_L(m) = \sin(m) - 0.1m - 0.1$. In this simple case, both the high- and low-fidelity models are very quick to simulate, and this case is only used to demonstrate the performance of multifidelity GP.

In the above approach, we only utilize data from two levels of model fidelity. If s levels of data $\{\mathbf{D}_t(\mathbf{M})\}_{t=1}^s$ sorted by increasing fidelity are available, we can readily extend the autoregressive scheme:

$$u_t(\mathbf{m}) = \rho_{t-1}u_{t-1}(\mathbf{m}) + \delta_t(\mathbf{m}), \quad t = 2, \dots, s, \quad (11)$$

where $\delta_t(\mathbf{m})$ is a Gaussian process independent of $\{u_{t-1}(\mathbf{m}), \dots, u_1(\mathbf{m})\}$, ρ_{t-1} is the cross-correlation coefficient. Moreover, we can consider complex, nonlinear relationship between $u_{t-1}(\mathbf{m})$ and $u_t(\mathbf{m})$ (Perdikaris et al., 2017):

$$u_t(\mathbf{m}) = q_{t-1}(u_{t-1}(\mathbf{m})) + \delta_t(\mathbf{m}), \quad t = 2, \dots, s, \quad (12)$$

where $q_{t-1}(\cdot)$ is a nonlinear mapping.

It is noted here that the autoregressive model considers the relationship between the high- and low-fidelity models, not the input-output relationship of the hydrologic system. In practice, the low-fidelity model should resemble and capture the right trend of the high-fidelity model, otherwise the low-fidelity model would be useless. In that case, using a linear autoregressive model is still applicable even in nonlinear problems. For the sake of clarity, the nonlinear mapping described above is not adopted in the present paper. For more details about GP and its construction with multifidelity data, one can refer to Kennedy and O'Hagan (2000), Parussini et al. (2017), Raissi et al. (2017), and Williams and Rasmussen (2006).

2.3. The Adaptive Multifidelity MCMC Algorithm

Generally, we can only afford a limited number of $f_H(\mathbf{m})$ evaluations. If the multifidelity system $\tilde{g}(\mathbf{m})$ is constructed over the whole prior distribution, its accuracy cannot be guaranteed. In MCMC simulations, our concern is the posterior distribution. Thus, it is crucial that $\tilde{g}(\mathbf{m})$ is accurate enough therein, but there is no need to ensure its accuracy elsewhere. Below we propose an AMF-MCMC algorithm which adaptively refines $\tilde{g}(\mathbf{m})$ over the posterior distribution and finally obtains an accurate estimate of $p(\mathbf{m}|\tilde{\mathbf{d}})$.

The AMF-MCMC algorithm first builds an initial multifidelity system $\tilde{g}_0(\mathbf{m})$ conditioned on N_L evaluations of $f_L(\mathbf{m})$ and N_H evaluations of $f_H(\mathbf{m})$, where N_H is usually a small integer and N_L is much larger than N_H . With $\tilde{g}_0(\mathbf{m})$, we can run DREAM_(ZS) to sufficiently explore the parameter space, which can be done very quickly. Here the variance of the GP system (σ_{gp}^2) is considered in the MCMC simulation by augmenting it with the variance of the measurement error (σ_{meas}^2); that is, $\sigma_{total}^2 = \sigma_{gp}^2 + \sigma_{meas}^2$, and using σ_{total}^2 in the likelihood function, $L(\mathbf{m}|\tilde{\mathbf{d}})$. Then we can draw two random samples, \mathbf{m}_H^p and \mathbf{m}_L^p , from the approximated

posterior, $\tilde{p}_0(\mathbf{m}|\tilde{\mathbf{d}})$, which are expected to be much closer to the posterior region than the prior samples. By utilizing the new data $f_H(\mathbf{m}_H^P)$ and $f_L(\mathbf{m}_L^P)$, the multifidelity GP system can be refined locally. This process will be further iterated $I_{\max} - 1$ times. Since it is very cheap to evaluate $f_L(\mathbf{m})$, we can also collect more samples from the approximated posterior for $f_L(\mathbf{m})$ simulations in GP refinement at each iteration. However, an increased number of training data will inevitably increase the computational cost in constructing the multifidelity GP system. To address this issue, one practical strategy is to remove some *bad* data according to the closeness of the training data to the measurements [e.g., the likelihood function defined in equation (3)].

In the AMF-MCMC algorithm, we gradually improve the accuracy of the multifidelity GP system by iteratively adding new training data from $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ simulations in the posterior distribution. At each iteration, the new training data are obtained by running MCMC based on the previous GP system to sufficiently explore the parameter space. As the GP error is considered, the MCMC simulation results will not be overconfident and biased (although the posterior will be wider). Based on both the new and old training data, we can obtain an updated GP system that is slightly more accurate in the posterior distribution. Finally, we can obtain a GP system that is locally accurate (i.e., with a negligible σ_{gp}^2) in the posterior region and an accurate estimate of $p(\mathbf{m}|\tilde{\mathbf{d}})$.

After each DREAM_(ZS) simulation, the thinned chain history \mathbf{Z} will be saved and used in the next DREAM_(ZS) simulation. Actually, this treatment tailors the $I_{\max} + 1$ $\tilde{g}(\mathbf{m})$ -based MCMC simulations into an integrated one, which is beneficial to better explore the parameter space. The complete scheme of the AMF-MCMC algorithm is given in Algorithm 1.

Algorithm 1 The adaptive multi-fidelity MCMC algorithm.

1. Draw N_L random samples from the prior distribution, $\mathbf{M}_L = [\mathbf{m}_1, \dots, \mathbf{m}_{N_L}]$, calculate $\mathbf{D}_L = [f_L(\mathbf{m}_1), \dots, f_L(\mathbf{m}_{N_L})]$.
2. Draw N_H random samples from the prior distribution, $\mathbf{M}_H = [\mathbf{m}_1, \dots, \mathbf{m}_{N_H}]$, calculate $\mathbf{D}_H = [f_H(\mathbf{m}_1), \dots, f_H(\mathbf{m}_{N_H})]$. Here $N_H \ll N_L$.
3. Build the initial GP system $\tilde{g}_0(\mathbf{m})$ with multi-fidelity simulation conditioned on $[\mathbf{M}_L \mathbf{M}_H]$ and $[\mathbf{D}_L \mathbf{D}_H]$.
4. Run MCMC with $\tilde{g}_0(\mathbf{m})$, obtain $\tilde{p}_0(\mathbf{m}|\tilde{\mathbf{d}})$.
5. **for** $i = 1, \dots, I_{\max}$ **do**
 Draw two random samples, \mathbf{m}_H^P and \mathbf{m}_L^P , from $\tilde{p}_{i-1}(\mathbf{m}|\tilde{\mathbf{d}})$, let $\mathbf{M}_H = [\mathbf{M}_H \mathbf{m}_H^P]$, $\mathbf{D}_H = [\mathbf{D}_H f_H(\mathbf{m}_H^P)]$, $\mathbf{M}_L = [\mathbf{M}_L \mathbf{m}_L^P]$, and $\mathbf{D}_L = [\mathbf{D}_L f_L(\mathbf{m}_L^P)]$.
 Build the GP system $\tilde{g}_i(\mathbf{m})$ conditioned on $[\mathbf{M}_L \mathbf{M}_H]$ and $[\mathbf{D}_L \mathbf{D}_H]$.
 Run MCMC with $\tilde{g}_i(\mathbf{m})$ and previous MCMC simulation results, obtain $\tilde{p}_i(\mathbf{m}|\tilde{\mathbf{d}})$.
end for
6. The posterior is approximated with $\tilde{p}_{I_{\max}}(\mathbf{m}|\tilde{\mathbf{d}})$.

3. Illustrative Examples

3.1. Example 1: Estimation of Soil Hydraulic and Thermal Parameters

In example 1, we first test the performance of the AMF-MCMC algorithm in estimating soil hydraulic and thermal parameters in a single-ring infiltration experiment (Nakhaei & Šimůnek, 2014). Here the processes of water flow and heat transport are considered. As shown in Figure 2, the flow domain is 100 cm \times 200 cm. The initial conditions for water content and temperature in the domain are 0.100 cm³/cm³ and 17.5 °C, respectively. The domain has three types of boundary conditions, that is, impervious condition at the two lateral boundaries and part of the upper boundary (represented by the green lines in Figure 2), free drainage condition at the lower boundary (represented by the blue line in Figure 2), and constant temperature (61.0 °C) and water content (0.430 cm³/cm³) conditions at part of the upper boundary (represented by the red line in Figure 2).

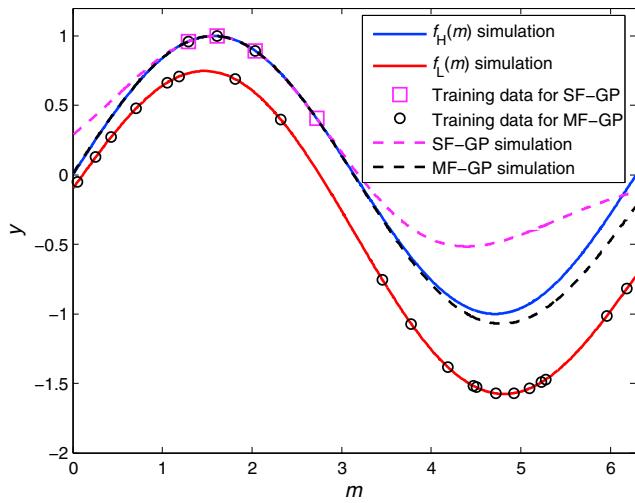


Figure 1. Comparison between the single-fidelity Gaussian process (SF-GP) simulation (magenta dashed curve) and the multifidelity Gaussian process (MF-GP) simulation (black dashed curve). Here the $f_H(m)$ and $f_L(m)$ simulations are represented by the blue curve and red curve, the training data for SF-GP and MF-GP are represented by the magenta squares and black circles, respectively.

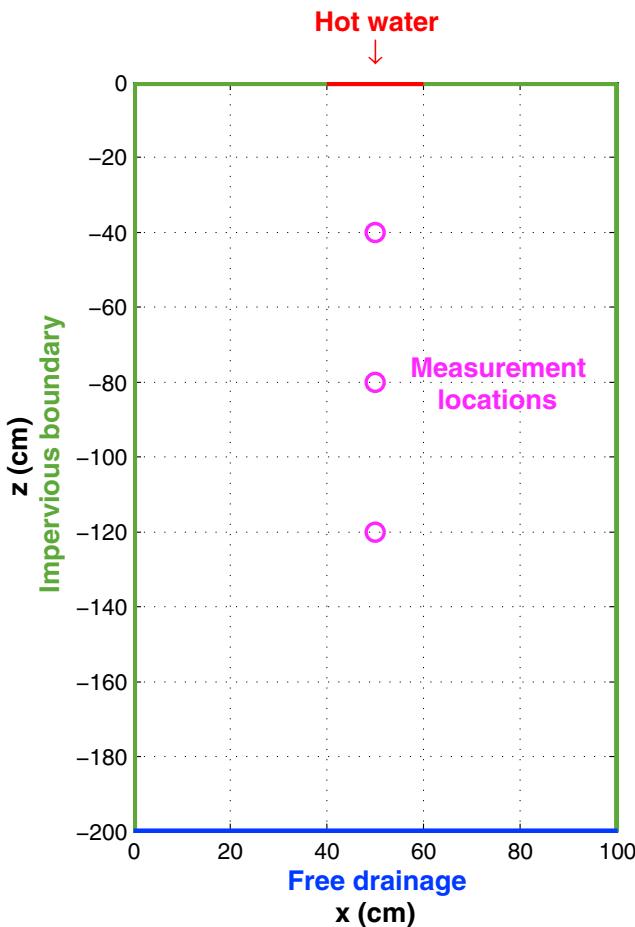


Figure 2. Flow domain for the first example.

With the initial and boundary conditions prescribed above, we can simulate unsaturated water flow in the domain by numerically solving the Richards' equation with HYDRUS-2D (Šimůnek et al., 2008):

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial x} \left[K(h) \frac{\partial h}{\partial x} \right] + \frac{\partial}{\partial z} \left[K(h) \frac{\partial h}{\partial z} + K(h) \right], \quad (13)$$

where $\theta[\text{L}^3\text{L}^{-3}]$ is volumetric water content of the soil; $t[\text{T}]$ is time; $x[\text{L}]$ and $z[\text{L}]$ are distances along the horizontal and vertical directions; $h[\text{L}]$ is pressure head; $K(h)[\text{LT}^{-1}]$ is hydraulic conductivity, which is a function of h (Mualem, 1976; Van Genuchten, 1980):

$$K(h) = K_s S_e^l \left[1 - \left(1 - S_e^{1/m} \right)^m \right]^2, \quad (14)$$

where $K_s[\text{LT}^{-1}]$ is saturated hydraulic conductivity, $S_e[-]$ is effective saturation:

$$S_e = \frac{\theta - \theta_r}{\theta_s - \theta_r} = \begin{cases} \frac{1}{(1 + |\alpha h|^n)^m} & h < 0, \\ 1 & h \geq 0 \end{cases}, \quad (15)$$

where θ_r and θ_s are residual and saturated water content [L^3L^{-3}]; $l[-]$ is a pore-connectivity parameter; $\alpha[\text{L}^{-1}]$, $n[-]$, and $m = (1 - 1/n)[-]$ are empirical shape parameters, respectively.

Based on the simulation results of unsaturated water flow, we can further simulate heat transport by numerically solving the following governing equation (Sophocleous, 1979) with HYDRUS-2D:

$$C(\theta) \frac{\partial T}{\partial t} = \frac{\partial}{\partial z} \left[\lambda_{xz}(\theta) \frac{\partial T}{\partial x} \right] - C_w q_z \frac{\partial T}{\partial z}, \quad (16)$$

where $C(\theta) = C_n \theta_n + C_o \theta_o + C_w \theta$ is volumetric heat capacity of soil [$\text{ML}^{-1}\text{T}^{-2}\text{K}^{-1}$], C_n , C_o , and C_w are volumetric heat capacities of solid phase, organic phase, and liquid phase [$\text{ML}^{-1}\text{T}^{-2}\text{K}^{-1}$], θ_n and θ_o are fraction of solid phase and organic phase [L^3L^{-3}], respectively; $T[\text{K}]$ is temperature; $\lambda_{xz}(\theta)[\text{ML}^{-3}\text{K}^{-1}]$ is apparent thermal conductivity:

$$\lambda_{xz}(\theta) = \lambda_x C_w |q| \delta_{xz} + (\lambda_z - \lambda_x) C_w \frac{q_x q_z}{|q|} + \lambda_0(\theta) \delta_{xz}, \quad (17)$$

where λ_x and λ_z are components of thermal dispersivity in the horizontal and vertical directions [L]; $q[\text{LT}^{-1}]$ is fluid flux density with absolute value $|q|$, and components in the horizontal and vertical directions, q_x and q_z , respectively; δ_{xz} is Kronecker delta function; $\lambda_0(\theta) = b_1 + b_2 \theta + b_3 \theta^{0.5}$ is thermal conductivity of soil in the absence of flow, where b_1 , b_2 , and b_3 are empirical parameters, [$\text{ML}^{-3}\text{K}^{-1}$]. In this example, the total simulation time is 10 hr. For more details about the model descriptions, one can refer to Nakhaei and Šimůnek (2014).

Here the unknown model parameters are α , n , K_s , b_1 , b_2 , and b_3 , which are assumed to be homogeneous in the domain and fit multivariate uniform prior distribution (Table 1), while other parameters are assumed to be known, as listed in Table 2. To infer the six unknown model parameters, we collect pressure head and temperature measurements at $t = 1, 2, \dots, 10$ hr at three locations denoted by the magenta circles in Figure 2. The measurements are generated from one set of true model parameters (Table 1) evaluated with $f_H(\mathbf{m})$ and perturbed with additive measurement errors, $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2)$. Here the standard deviation of the measurement errors

Table 1

Prior Ranges and True Values of Unknown Model Parameters in the First Example

Parameter	Prior range	True value
$\alpha[\text{cm}^{-1}]$	[0.0190 – 0.0930]	0.0387
$n[-]$	[1.360 – 2.370]	2.210
$K_s[\text{cm}/\text{hr}]$	[4.828 – 11.404]	6.759
$b_1[\text{kg} \cdot \text{cm} \cdot \text{hr}^{-3} \cdot \text{K}^{-1}]$	[2.179×10^{12} to 4.857×10^{12}]	2.948×10^{12}
$b_2[\text{kg} \cdot \text{cm} \cdot \text{hr}^{-3} \cdot \text{K}^{-1}]$	[4.778×10^{11} to 2.426×10^{12}]	2.118×10^{12}
$b_3[\text{kg} \cdot \text{cm} \cdot \text{hr}^{-3} \cdot \text{K}^{-1}]$	[2.174×10^{12} to 5.184×10^{12}]	2.972×10^{12}

for pressure head is $\sigma_h = 1 \text{ cm}$, the standard deviation of the measurement errors for temperature is $\sigma_T = 0.5^\circ\text{C}$. Then we use the Gaussian likelihood function defined in equation (3) to evaluate the goodness of fit between the model simulation and measurement data.

In this example, the high- and low-fidelity models are constructed with different levels of discretization. In $f_H(\mathbf{m})$, the flow domain is evenly discretized into 41×41 grids, while in $f_L(\mathbf{m})$, there are 21×21 grids. In a Dell Precision T7610 workstation (Intel Xeon Processor E5-2680 v2 @ 2.80GHz; 512GB RAM; Windows 10 Pro., 64 Bit), the average time of evaluating $f_H(\mathbf{m})$ once is about 54 s, while the average time of evaluating $f_L(\mathbf{m})$ once

is about 5.6 s. We admit here that the high-fidelity model used in this test case is not actually very time consuming, so that we can implement the high-fidelity MCMC simulation to obtain the reference results to verify the performance of the AMF-MCMC algorithm. As shown in Figure 3, although $f_L(\mathbf{m})$ can capture the main trend of $f_H(\mathbf{m})$, systematic errors do exist.

We first run MCMC with $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$, respectively to approximate the posterior. Here the two MCMC simulations have four parallel chains with 4,000 iterations, which means 16,000 model evaluations for each simulation. Using the \hat{R} -statistic proposed in Brooks & Gelman (1998) and Gelman & Rubin (1992), we can monitor the convergence of the $f_H(\mathbf{m})$ - and $f_L(\mathbf{m})$ -MCMC simulations. In each panel of Figure 4, traces of the \hat{R} -statistic of different parameters are coded with different colors. The red dashed line in both panels demarcates the threshold of 1.2 below which the chains are assumed to have converged to a stationary distribution. As shown in Figure 5, the $f_H(\mathbf{m})$ -MCMC simulation can identify the unknown model parameters properly, while the estimation results of the $f_L(\mathbf{m})$ -MCMC simulation are significantly biased, especially for the first three parameters. Thus, although using a low-fidelity model in the MCMC simulation can gain computational efficiency, the estimation accuracy cannot be guaranteed.

We then run the AMF-MCMC algorithm with the same set of measurements to approximate the posterior. Here the initial number of $f_L(\mathbf{m})$ evaluations is $N_L = 200$, and the initial number of $f_H(\mathbf{m})$ evaluations is $N_H = 30$. These initial samples are randomly drawn from the prior distribution. Conditioned on these multifidelity data, we build the initial GP system, $\tilde{g}_0(\mathbf{m})$, based on which the MCMC simulation (four parallel chains with 4,000 iterations) can be implemented very quickly. From the approximated posterior, $\tilde{p}_0(\mathbf{m}|\tilde{\mathbf{d}})$, we can draw two random parameter samples, \mathbf{m}_H^p and \mathbf{m}_L^p , which are expected to be much closer to the posterior region than the prior samples. Then we can add $f_H(\mathbf{m}_H^p)$ and $f_L(\mathbf{m}_L^p)$ to the existing multifidelity training data to refine the GP system locally. The coupled process of GP-based MCMC simulation and GP system refinement is further repeated 69 times (i.e., $I_{\max} = 70$). In Figure 6, we plot the evolution of the variance of the multifidelity GP output [i.e., $\hat{\sigma}_{gp}(h)$ and $\hat{\sigma}_{gp}(T)$] and the root-mean-square error (RMSE) between the multifidelity GP output and the high-fidelity output [i.e., RMSE(h) and RMSE(T)] averaged over 400 posterior samples. It is found that for both hydraulic head and temperature outputs, the accuracy of the multifidelity GP system will improve with the iteration. Finally, the multifidelity GP system will be rather accurate in the posterior region.

As shown in Figure 7, the successively added new parameter samples (blue dots) for $f_H(\mathbf{m})$ evaluations gradually approach to the true values (black crosses), which are the basis of a locally accurate GP system.

Based on this GP system, we can obtain a rather accurate approximation of $p(\mathbf{m}|\tilde{\mathbf{d}})$. As shown in Figure 8, the marginal posterior probability density functions obtained by the $f_H(\mathbf{m})$ -MCMC algorithm and the AMF-MCMC algorithm are almost identical, which confirms the accuracy of the AMF-MCMC algorithm.

In the AMF-MCMC algorithm, the total numbers of $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ evaluations are 100 and 270, respectively. With respect to the number of model evaluations, the AMF-MCMC algorithm is much more efficient than both the $f_H(\mathbf{m})$ - and $f_L(\mathbf{m})$ -based MCMC simulations. It is noted here that, when building a GP system, the training data of model parameters are a matrix of $N_m \times N_{tr}$, while the training data of model outputs should be a vector of $1 \times N_{tr}$, where N_{tr} is the total number of training data. It means that

Table 2

Values of Known Model Parameters in the First Example

Parameter	Value	Parameter	Value
$\theta_r[\text{cm}^3 \text{cm}^{-3}]$	0.041	$l[-]$	0.500
$\theta_s[\text{cm}^3 \text{cm}^{-3}]$	0.430	$C_w[\text{J} \cdot \text{cm}^{-3} \cdot \text{K}^{-1}]$	4.180
$\theta_n[\text{cm}^3 \text{cm}^{-3}]$	0.600	$C_n[\text{J} \cdot \text{cm}^{-3} \cdot \text{K}^{-1}]$	1.920
$\theta_o[\text{cm}^3 \text{cm}^{-3}]$	0.001	$C_o[\text{J} \cdot \text{cm}^{-3} \cdot \text{K}^{-1}]$	2.510
$\lambda_x[\text{cm}]$	0.200	$\lambda_y[\text{cm}]$	2.000

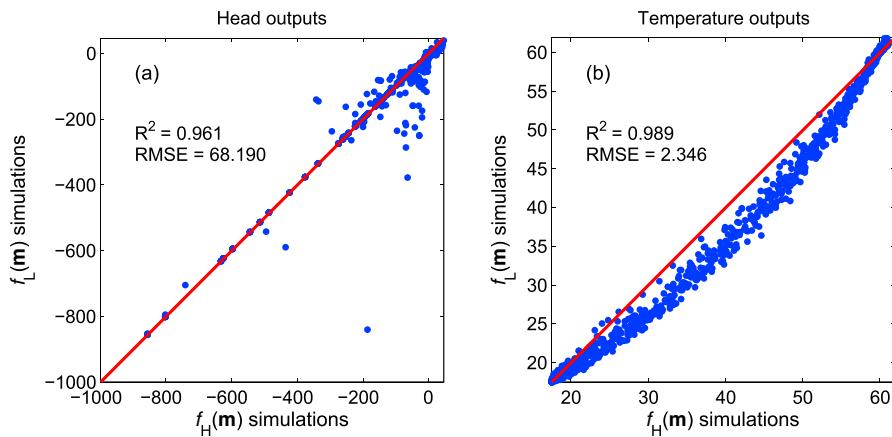


Figure 3. Comparison of simulated (a) head and (b) temperature outputs between $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$.

when the model outputs are N_d dimensional, we have to build N_d GP systems for each of the N_d model outputs separately. When we run the high- or low-fidelity model to acquire the training data, we can obtain the N_d model outputs at the same time. So the total number of function evaluations that are used to build GPs for all observation data points is still N_r . Nevertheless, the time needed by the multifidelity GP system constructions and GP-based MCMC simulations should not be neglected, especially when the number of model outputs N_d is large. To improve the efficiency of the AMF-MCMC algorithm, we can build the N_d GPs in parallel. In our simulations, there are 20 cores available, which can be utilized to greatly accelerate the simulation of the AMF-MCMC algorithm.

In the above simulation, we only add one set of $f_H(\mathbf{m})$ data and one set of $f_L(\mathbf{m})$ data at each iteration. As stated in section 2.3, we can also add more than one set of new data once. In Figure 9, we compare the finally obtained variance of the multifidelity GP output [i.e., $\hat{\sigma}_{gp}(h)$ and $\hat{\sigma}_{gp}(T)$] and RMSE between the multifidelity GP output and the high-fidelity output [i.e., RMSE(h) and RMSE(T)] when adding different numbers of new training data once. Here the iteration numbers are set as 70, 35, 14, 7, 5, and 4 for adding 1, 2, 5, 10, 15, and 20 sets of $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ training data at each iteration, respectively, to make sure that the total numbers of added data sets are roughly equal. It is clear that in this case, adding one set of $f_H(\mathbf{m})$ data and one set of $f_L(\mathbf{m})$ data once can bring about a more accurate multifidelity system in the posterior region.

In our previous paper (Zhang et al., 2016), we proposed an efficient method to estimate hydrologic model parameters by combining MCMC simulations with a GP surrogate adaptively refined over the posterior

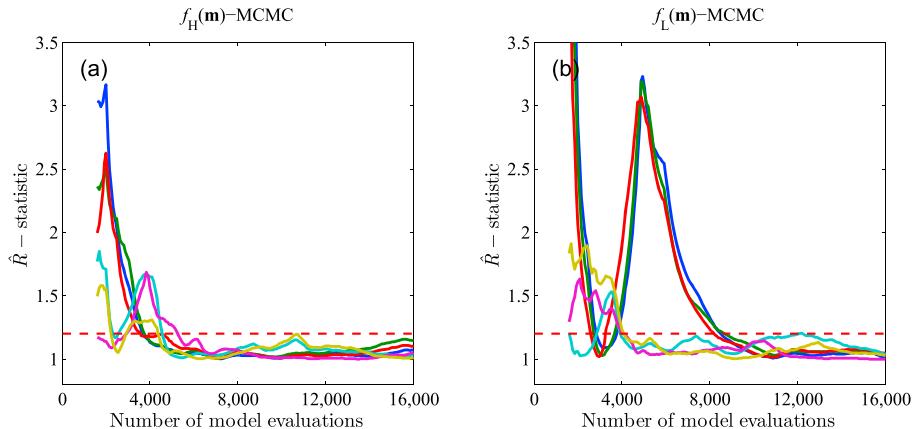


Figure 4. Trace plots of the \hat{R} -statistic of the six model parameters in (a) $f_H(\mathbf{m})$ -MCMC simulation and (b) $f_L(\mathbf{m})$ -MCMC simulation. The threshold of 1.2 for convergence diagnosis is represented by the red dashed lines. MCMC = Markov chain Monte Carlo.

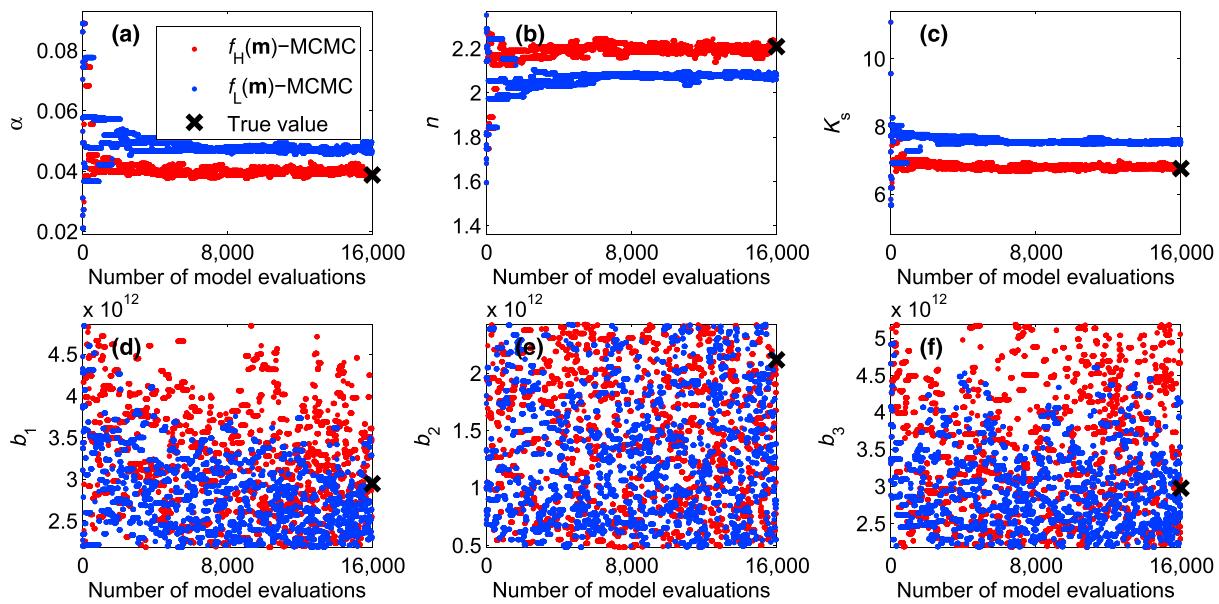


Figure 5. Trace plots of model parameters obtained by the $f_H(\mathbf{m})$ -MCMC algorithm (red dots) and the $f_L(\mathbf{m})$ -MCMC algorithm (blue dots) in the first example. The true values are represented by the black crosses. MCMC = Markov chain Monte Carlo.

distribution. Here we call that method the adaptive Gaussian process MCMC (AGP-MCMC) algorithm. It is obvious that the AMF-MCMC algorithm proposed in the present paper is an extension of the AGP-MCMC algorithm through introducing the adaptive multifidelity simulation. As the AMF-MCMC algorithm takes advantage of the correlation relationship between $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$, it generally requires fewer $f_H(\mathbf{m})$ evaluations than the AGP-MCMC algorithm. In other words, with similar computational cost, the multifidelity GP system helps to better explore the parameter space compared to the single-fidelity GP system. To demonstrate this point, we further compare the performances of the two algorithms in this example when limited numbers of $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ evaluations are affordable. In the AMF-MCMC algorithm, the initial number of $f_L(\mathbf{m})$ evaluations is $N_L = 60$, the initial number of $f_H(\mathbf{m})$ evaluations is $N_H = 10$, and an extra number of $I_{\max} = 40$ $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ evaluations are called to adaptively refine the multifidelity GP system in the posterior region.

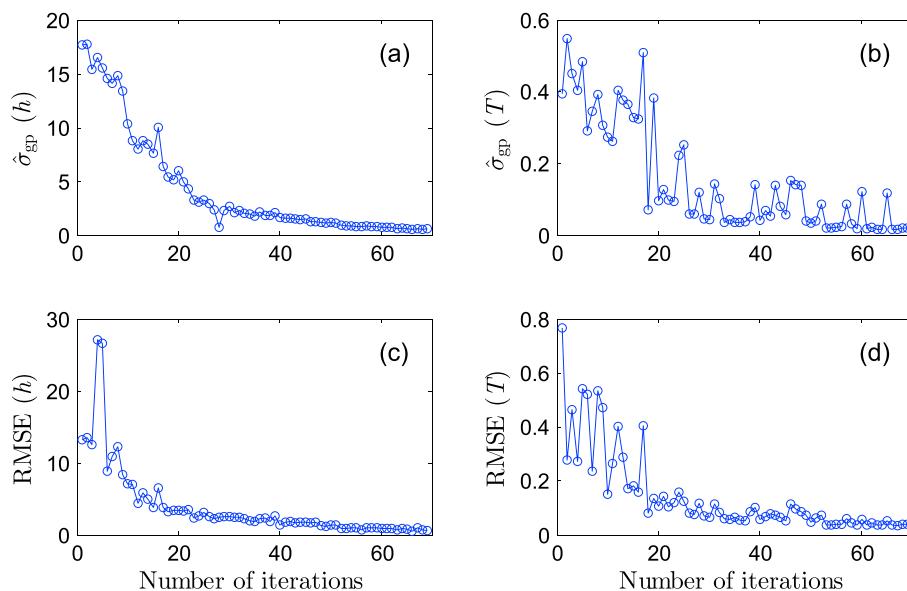


Figure 6. Evolution of (a, b) variance of the multifidelity GP output and (c, d) RMSE between the multifidelity GP output and the high-fidelity output averaged over 400 posterior samples. GP = Gaussian process; RMSE = root-mean-square error.

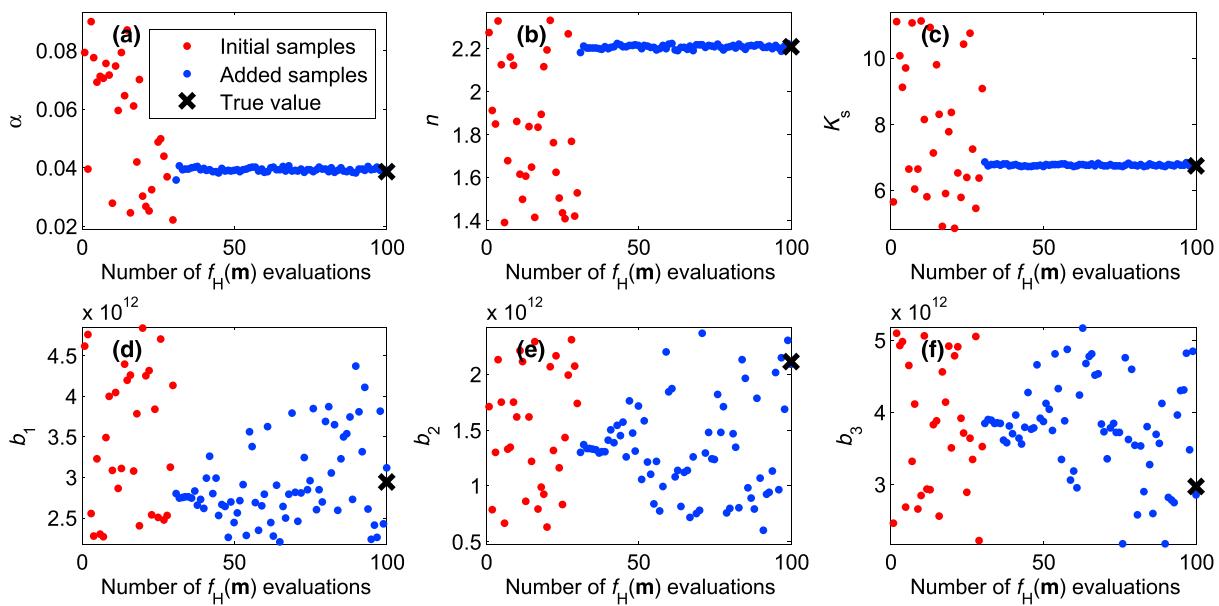


Figure 7. N_H initial parameter samples (red dots) and l_{\max} successively added parameter samples (blue dots) for $f_H(\mathbf{m})$ evaluations in the AMF-MCMC algorithm. The true values are represented by the black crosses. AMF-MCMC = adaptive multifidelity-Markov chain Monte Carlo.

In the AGP-MCMC algorithm, we first evaluate $N_H = 20$ prior samples with $f_H(\mathbf{m})$. Then we further evaluate $l_{\max} = 40$ more parameter samples with $f_H(\mathbf{m})$ that are gradually approaching to the posterior region. Since the simulation time of 100 $f_L(\mathbf{m})$ evaluations is approximately equal to the simulation time of 10 $f_H(\mathbf{m})$ evaluations, the computational costs in model evaluations for AMF-MCMC and AGP-MCMC are roughly equal. As shown in Figures 10a–10c, the adaptively added parameter samples by the AMF-MCMC algorithm converge faster to the true values. For the AGP-MCMC algorithm, only 60 $f_H(\mathbf{m})$ evaluations in total are still not enough, especially for the parameters b_1 and b_3 .

As shown in Figure 11, the finally estimated posterior probability density functions of the unknown model parameters by the AMF-MCMC algorithm (blue dashed curves) are overall much closer to the reference

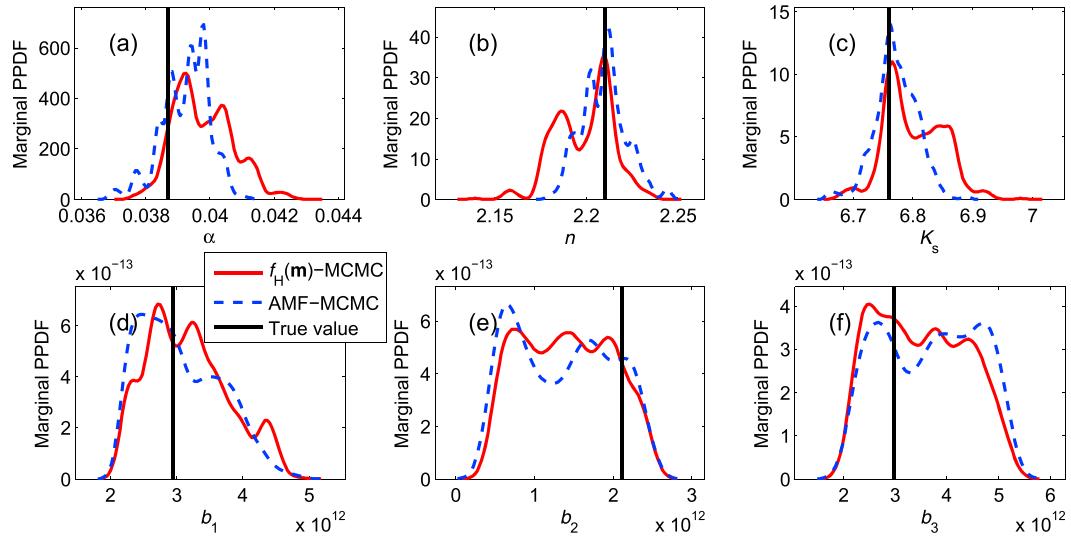


Figure 8. Marginal posterior probability density functions obtained by the $f_H(\mathbf{m})$ -MCMC algorithm (red curves) and the AMF-MCMC algorithm (blue dashed curves). The true values are represented by the black vertical lines. AMF-MCMC = adaptive multifidelity-Markov chain Monte Carlo.

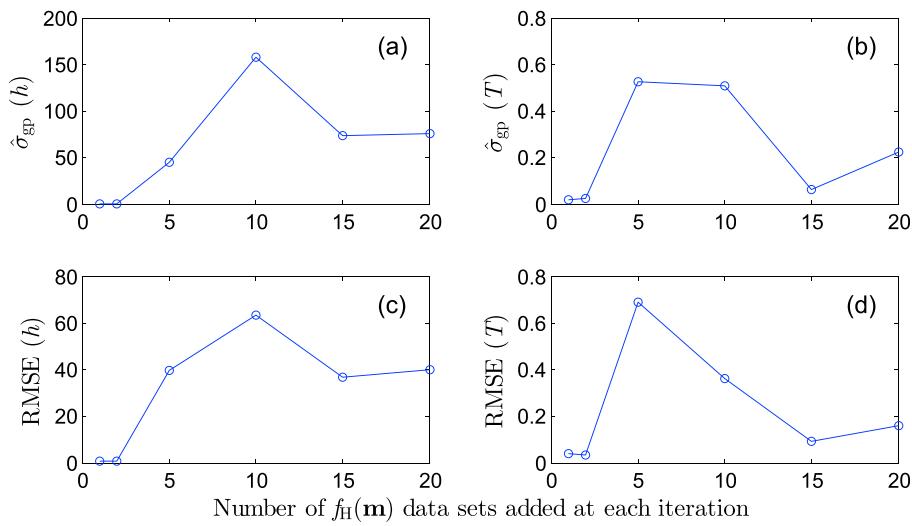


Figure 9. Finally obtained (a, b) variance of the multifidelity GP output and (c, d) RMSE between the multifidelity GP output and the high-fidelity output when adding different numbers of new training data sets at each iteration. GP = Gaussian process; RMSE = root-mean-square error.

results (red curves) than the AGP-MCMC algorithm (magenta dash-dotted curves). Therefore, it is important to leverage the multifidelity system based on a large number of $f_L(\mathbf{m})$ simulations to sufficiently explore the parameter space especially at the early stage of the GP-based MCMC simulations. In practical applications, to guarantee the estimation accuracy of the AMF-MCMC algorithm, more evaluations of $f_H(\mathbf{m})$ are suggested. For more details about the AGP-MCMC algorithm, one can refer to Zhang et al. (2016) and a more recent work by other researchers (Gong & Duan, 2017).

3.2. Example 2: Contaminant Source Identification With Multimodal Posterior

In example 2, we further test the performance of the AMF-MCMC algorithm in solving an inverse problem with multimodal posterior distribution. Here we consider the processes of steady state saturated

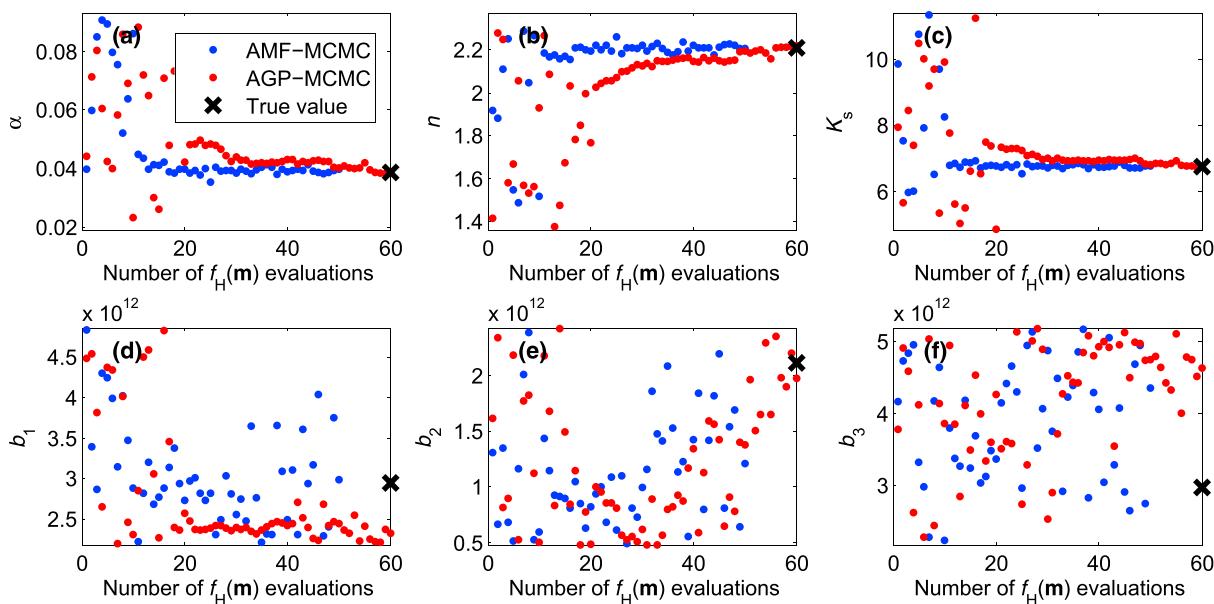


Figure 10. Trace plots of model parameters evaluated by $f_H(\mathbf{m})$ in the AMF-MCMC algorithm (blue dots) and the AGP-MCMC algorithm (red dots). The true values are represented by the black crosses. AMF-MCMC = adaptive multifidelity-Markov chain Monte Carlo.

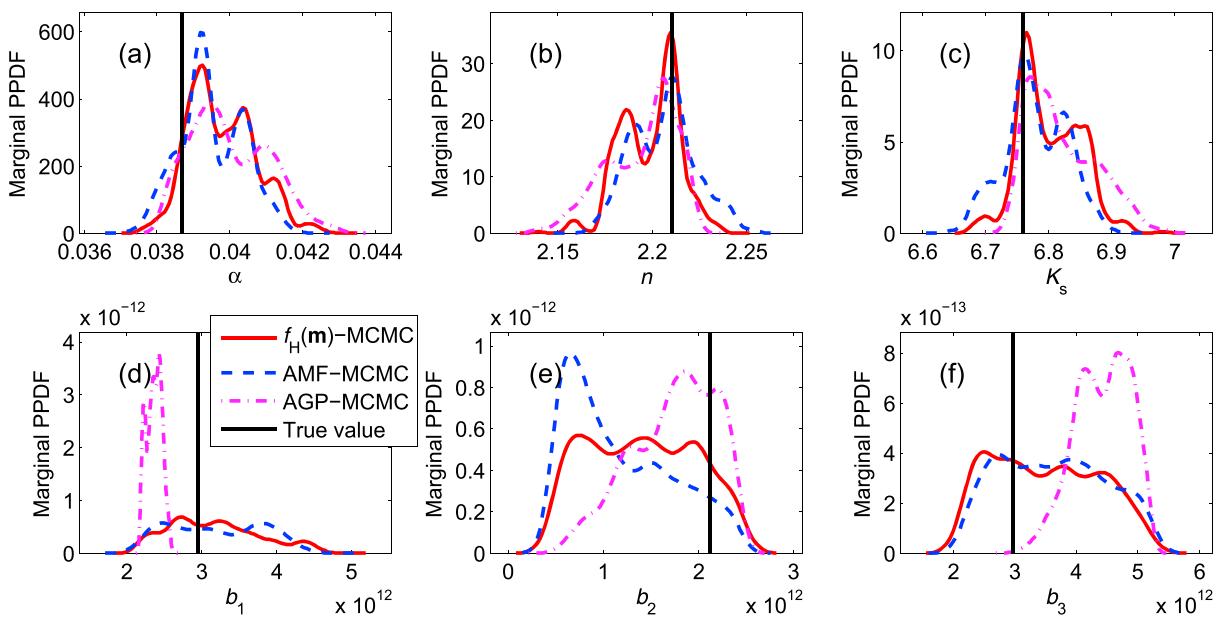


Figure 11. Marginal posterior probability density functions obtained by the $f_H(\mathbf{m})$ -MCMC algorithm (red curves), the AMF-MCMC algorithm (blue dashed curves), and the AGP-MCMC algorithm (magenta dash-dotted curves), respectively. The true values are represented by the black vertical lines. AMF-MCMC = adaptive multifidelity-Markov chain Monte Carlo.

groundwater flow and contaminant transport. As shown in Figure 12, the flow domain is $20[\text{L}] \times 10[\text{L}]$. The upper and lower sides are no-flow boundaries, the left ($h = 12[\text{L}]$) and right ($h = 11[\text{L}]$) sides are constant-head boundaries, respectively. At the initial time, hydraulic heads in the domain are all $11[\text{L}]$ except for the left boundary ($12[\text{L}]$).

Given the above initial and boundary conditions, we can obtain the flow field through numerically solving the following governing equations with MODFLOW (Harbaugh et al., 2000):

$$\frac{\partial}{\partial x_i} \left(K_i \frac{\partial h}{\partial x_i} \right) = 0, \quad (18)$$

and

$$v_i = -\frac{K_i}{\theta} \frac{\partial h}{\partial x_i}, \quad (19)$$

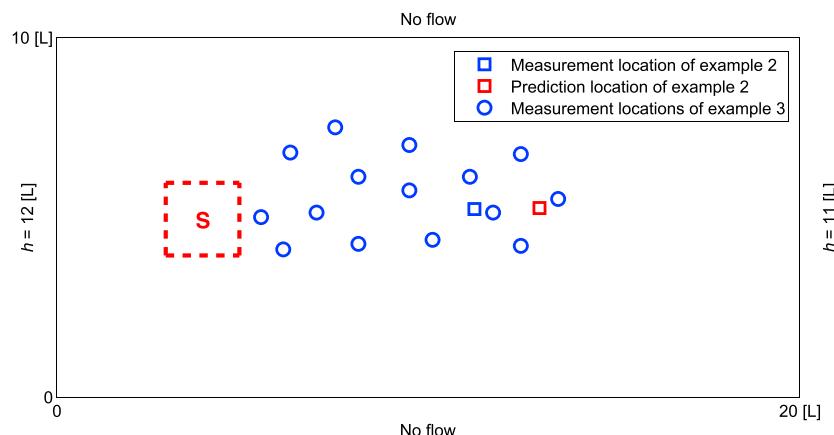


Figure 12. Flow domain for the second and third examples.

Table 3

Prior Ranges and True Values of Unknown Model Parameters in the Second Example

Parameter	Range	True value
$x_s[\text{L}]$	[3–5]	3.854
$y_s[\text{L}]$	[3–7]	5.999
$S_s[\text{MT}^{-1}]$	[10–13]	11.044
$t_{\text{on}}[\text{T}]$	[3–5]	4.897
$t_{\text{off}}[\text{T}]$	[9–11]	9.075

where $x_i[\text{L}]$, $K_i[\text{LT}^{-1}]$, and $v_i[\text{LT}^{-1}]$ signify distance, hydraulic conductivity, and pore water velocity along the respective Cartesian coordinate axis, for $i = 1, 2$; $h[\text{L}]$ is hydraulic head; $\theta[-]$ is porosity of the aquifer. Here $K = 8[\text{LT}^{-1}]$ and $\theta = 0.25$ are known beforehand. In the steady state flow field, some amount of contaminant is released from a point source located somewhere in the red dashed rectangle depicted in Figure 12. The contaminant source is characterized by five parameters, that is, location, $(x_s, y_s)[\text{L}]$, source strength measured by mass loading rate, $S_s[\text{MT}^{-1}]$, start time of contaminant release, $t_{\text{on}}[\text{T}]$, and end time of the release, $t_{\text{off}}[\text{T}]$. Then we can obtain contaminant concentration $C[\text{ML}^{-3}]$ at different times and locations through numerically solving the following advection-dispersion equation with MT3DMS (Zheng & Wang, 1999):

$$\frac{\partial(\theta C)}{\partial t} = \frac{\partial}{\partial x_i} \left(\theta D_{ij} \frac{\partial C}{\partial x_j} \right) - \frac{\partial}{\partial x_i} (\theta v_i C) + q_s C_s, \quad (20)$$

where $t[\text{T}]$ is time; $q_s[\text{T}^{-1}]$ and $C_s[\text{ML}^{-3}]$ are volumetric flow rate per unit volume of the aquifer and concentration of the source, respectively; $D_{ij}[\text{L}^2\text{T}^{-1}]$ are hydrodynamic dispersion coefficient tensors:

$$\begin{cases} D_{11} = (\alpha_L v_1^2 + \alpha_T v_2^2) / |\mathbf{v}|, \\ D_{22} = (\alpha_L v_2^2 + \alpha_T v_1^2) / |\mathbf{v}|, \\ D_{12} = D_{21} = (\alpha_L - \alpha_T) v_1 v_2 / |\mathbf{v}|, \end{cases} \quad (21)$$

where $\alpha_L = 0.3[\text{L}]$ is longitudinal dispersivity, $\alpha_T = 0.03[\text{L}]$ is transverse dispersivity, and $|\mathbf{v}| = \sqrt{v_1^2 + v_2^2}$ is magnitude of the velocity.

In this example, the unknown model parameters to be estimated are the five contaminant source parameters. Here we assume that our prior knowledge about them are rather limited and thus represented by a multivariate uniform distribution (Table 3). To infer the five unknown parameters, we collect concentration measurements at $t = 6, 8, 10, 12, 14[\text{T}]$ at a well, denoted by the blue square in Figure 12. Here the measurements are generated from one set of true model parameters (Table 3) evaluated with $f_H(\mathbf{m})$ and perturbed with additive measurement errors that fit $\mathcal{N}(0, 0.01^2)$.

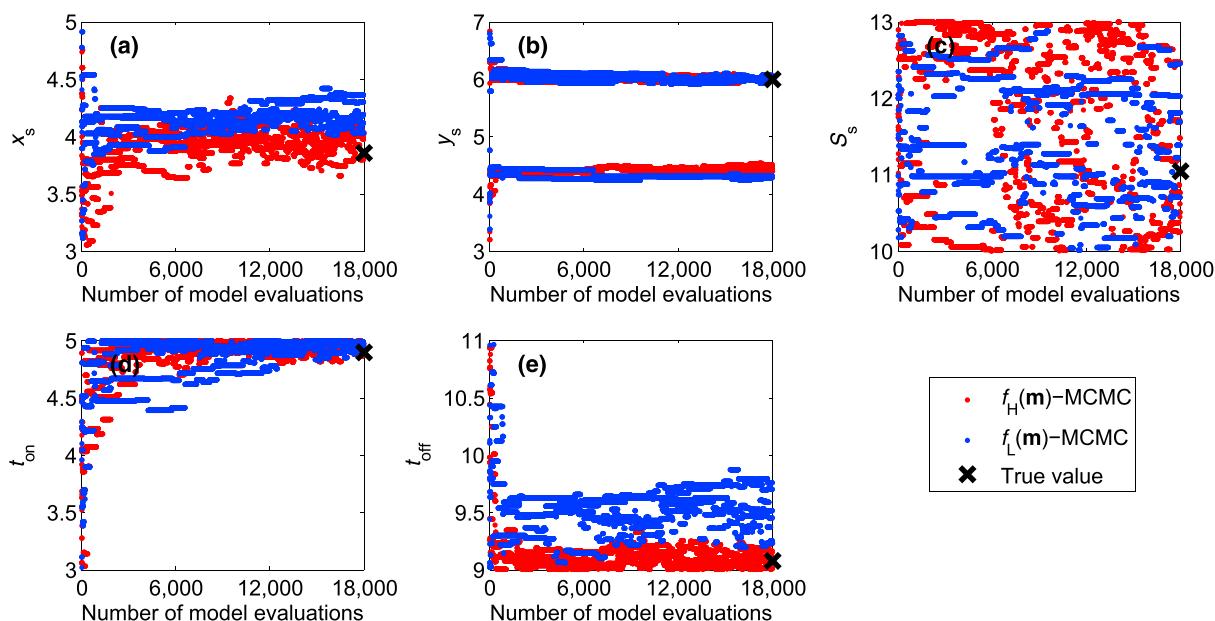


Figure 13. Trace plots of model parameters obtained by the $f_H(\mathbf{m})$ -MCMC algorithm (red dots) and the $f_L(\mathbf{m})$ -MCMC algorithm (blue dots) in the second example. The true values are represented by the black crosses. MCMC = Markov chain Monte Carlo.

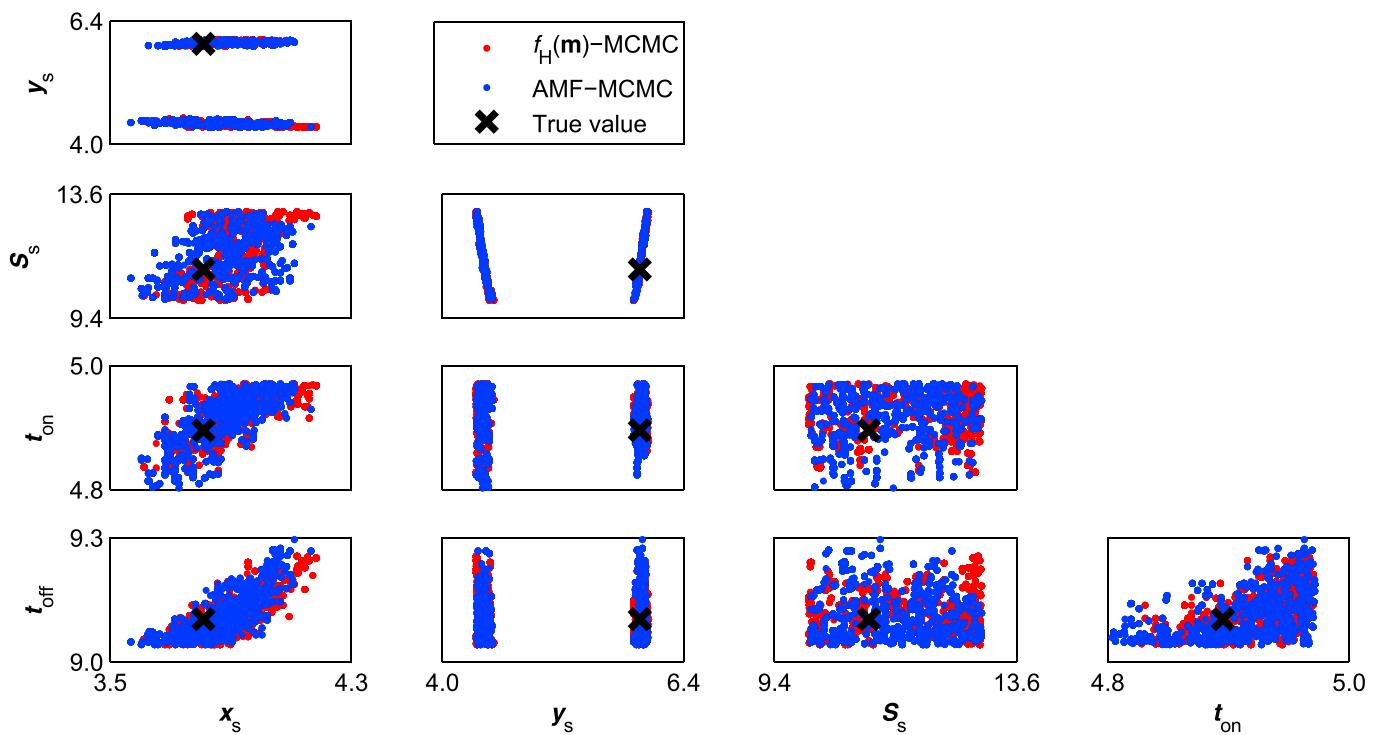


Figure 14. Bivariate scatter plots of posterior parameter samples obtained by the $f_H(\mathbf{m})$ -MCMC algorithm (red dots) and the AMF-MCMC algorithm (blue dots) in the second example. The true values are represented by the black crosses. AMF-MCMC = adaptive multifidelity-Markov chain Monte Carlo.

In this example, the high-fidelity model $f_H(\mathbf{m})$ is the coupled numerical model built with MODFLOW and MT3DMS. While the low-fidelity model $f_L(\mathbf{m})$ is built with a data-driven surrogate, that is, the adaptive sparse grid interpolation method proposed by Klimke and Wohlmuth (2005), based on 41 evaluations of $f_H(\mathbf{m})$. We compare the simulation results between $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ with 50 random parameter samples drawn from the prior distribution and obtain $R^2 = 0.949$. It should be pointed out here that other low-fidelity models, for example, a numerical model with a coarser discretization, can also be utilized. Considering that data-driven surrogates are widely used in hydrologic science (Asher et al., 2015; Razavi et al., 2012), and many times in MCMC simulations (Elsheikh et al., 2014; Laloy et al., 2013; L. Zeng et al., 2012; X. K. Zeng et al., 2016), here we test the applicability of data-driven surrogates in the AMF-MCMC algorithm. Then we run MCMC with

$f_H(\mathbf{m})$ and $f_L(\mathbf{m})$, respectively, to approximate the posterior (six parallel chains and 3,000 iterations). As shown in Figure 13, the results obtained by the $f_L(\mathbf{m})$ -MCMC simulation are significantly biased due to the approximation errors, especially for x_s and t_{off} .

With the same set of measurements, we further run the AMF-MCMC algorithm to approximate the posterior. Here the initial number of $f_L(\mathbf{m})$ evaluations is $N_L = 300$, and the initial number of $f_H(\mathbf{m})$ evaluations is $N_H = 30$. Then we successively add another $l_{max} = 50$ parameter samples for both $f_H(\mathbf{m})$ and $f_L(\mathbf{m})$ evaluations to refine the GP system over the posterior distribution. Finally, we can obtain a rather accurate approximation of $p(\mathbf{m}|\tilde{\mathbf{d}})$.

As shown in Figure 14, the bivariate scatter plots of the posterior samples obtained by the $f_H(\mathbf{m})$ -MCMC algorithm and the AMF-MCMC algorithm are almost identical, which indicates the accuracy of the AMF-MCMC algorithm. Moreover, the bimodality of y_s is well identified by both algorithms.

Here we also test how the parametric uncertainty affects the credible intervals of the model simulation and prediction for the different

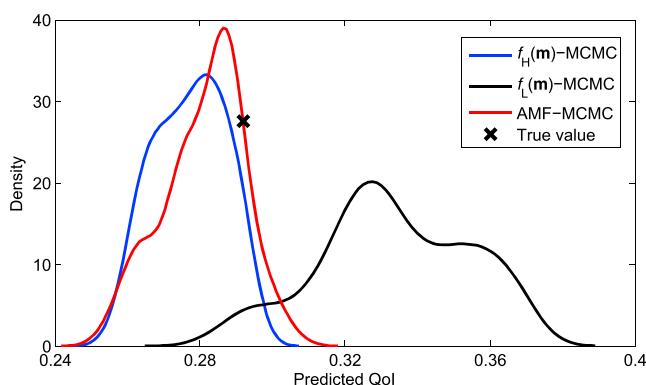


Figure 15. Predictive uncertainty of the Qol based on estimation results of the $f_H(\mathbf{m})$ -MCMC simulation (blue curve), the $f_L(\mathbf{m})$ -MCMC simulation (black curve), and the AMF-MCMC simulation (red curve), respectively. The true value of the Qol is represented by the black cross. MCMC = Markov chain Monte Carlo.

Table 4

Prior Ranges and True Values of Contaminant Source Parameters in the Third Example

Parameter	Range	True value
x_s [L]	[3–5]	4.033
y_s [L]	[4–6]	5.405
s_1 [MT^{-1}]	[0–8]	1.229
s_2 [MT^{-1}]	[0–8]	7.628
s_3 [MT^{-1}]	[0–8]	4.327
s_4 [MT^{-1}]	[0–8]	5.438
s_5 [MT^{-1}]	[0–8]	0.293
s_6 [MT^{-1}]	[0–8]	6.474

MCMC approaches. In this case, the quantity of interest (QoI) is the concentration at a well denoted by the red square in Figure 12 at a future time $t = 16[T]$. To quantify the predictive uncertainty of the QoI, we further run $f_H(\mathbf{m})$ to the future time given posterior samples obtained by the $f_H(\mathbf{m})$ -MCMC simulation, the $f_L(\mathbf{m})$ -MCMC simulation, and the AMF-MCMC simulation, respectively. As shown in Figure 15, the AMF-MCMC simulation (red curve) can obtain similar uncertain range of the QoI as the $f_H(\mathbf{m})$ -MCMC simulation (blue curve). However, using the posterior samples of the $f_L(\mathbf{m})$ -MCMC simulation can result in a wider and more biased prediction of the QoI (black curve).

3.3. Example 3: Contaminant Source Identification With 28 Unknown Parameters

In the first two cases, the numbers of unknown parameters are relatively small. To demonstrate the performance of the AMF-MCMC algorithm in high-dimensional problems, we further test a third numerical case that has 28 unknown parameters. This case is an extension of the second one by considering a more complex contaminant source and heterogeneous conductivity field.

Here the contaminant source is characterized by eight parameters, that is, the source location (x_s, y_s) and the source strengths s_i during $t = i : i + 1[T]$, for $i = 1, \dots, 6$. Here we assume that our prior knowledge about the eight source parameters are rather limited and represented by a multivariate uniform distribution (Table 4).

The log-transformed conductivity $Y = \ln K$ is assumed to be spatially correlated in the following form:

$$C_Y(x_1, y_1; x_2, y_2) = \sigma_Y^2 \exp\left(-\frac{|x_1 - x_2|}{\lambda_x} - \frac{|y_1 - y_2|}{\lambda_y}\right), \quad (22)$$

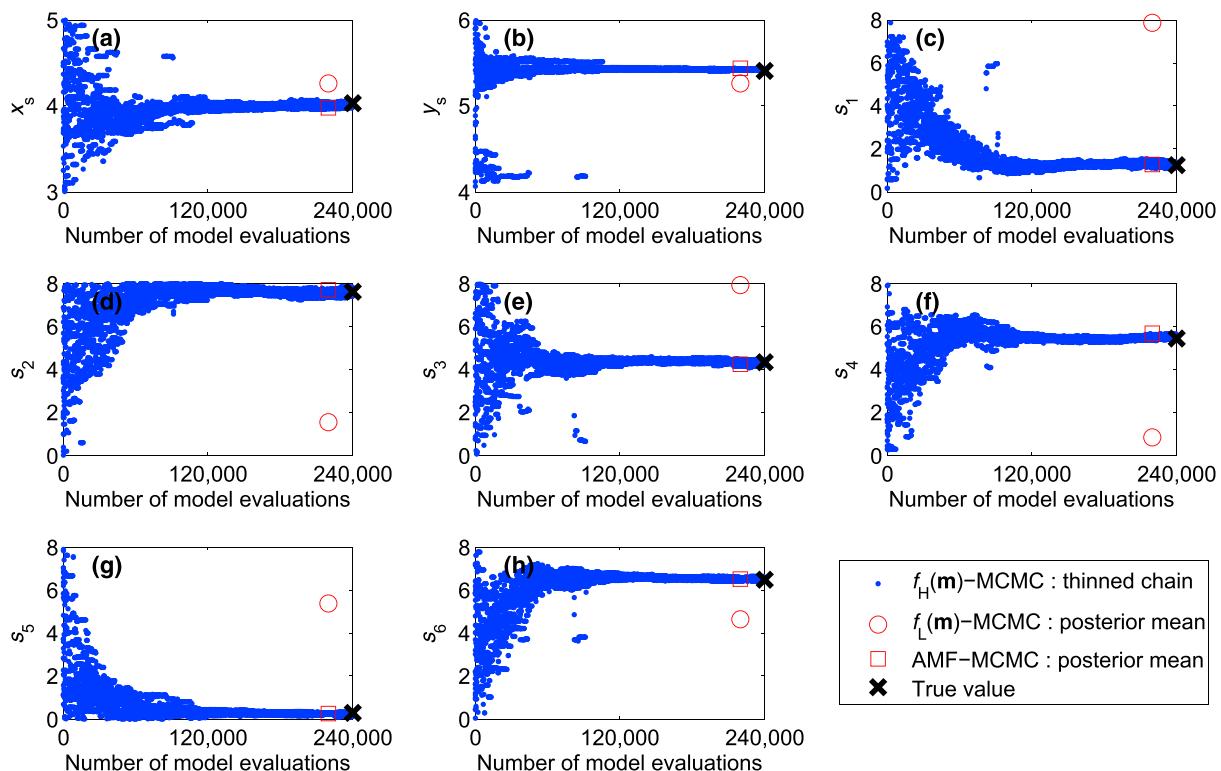


Figure 16. Trace plots of the $f_H(\mathbf{m})$ -MCMC simulation (blue dots), posterior mean estimates obtained by the $f_L(\mathbf{m})$ -MCMC simulation (red circles), and the AMF-MCMC simulation (red squares) of the eight contaminant source parameters. AMF-MCMC = adaptive multifidelity-Markov chain Monte Carlo.

where (x_1, y_1) and (x_2, y_2) are two arbitrary locations in the flow domain, $\sigma_Y^2 = 0.4$ is the variance of the Y field, $\lambda_x = 10[\text{L}]$ and $\lambda_y = 5[\text{L}]$ are the correlation lengths in the x and y directions, respectively. To reduce the dimensionality of the Y field, we adopt the Karhunen-Loëve (KL) expansion (Zhang & Lu, 2004) in this case:

$$Y(\mathbf{x}) \approx \bar{Y}(\mathbf{x}) + \sum_{i=1}^{N_{KL}} \sqrt{\tau_i} s_i(\mathbf{x}) \xi_i, \quad (23)$$

where $\bar{Y} = 2$ is the mean value of the Y field, τ_i and $s_i(\mathbf{x})$ are the eigenvalues and eigenfunctions of the correlation function defined in equation (22), $\xi_i \sim \mathcal{N}(0, 1)$ are independent Gaussian random variables, for $i = 1, \dots, N_{KL}$, respectively. Here $N_{KL} = 20$ is the number of truncated KL terms, which can preserve about 88.3% of the field variance. Then the unknown model parameters for the heterogeneous Y field are transformed to the 20 KL terms, that is, ξ_i , for $i = 1, \dots, 20$.

To infer the 28 unknown parameters for the contaminant source and conductivity field, we collect measurements of concentration and hydraulic head at wells denoted by the blue circles in Figure 12. The concentration measurements are collected at $t = 4, 6, 8, 10, 12[\text{T}]$, and the hydraulic head measurements are collected only once. The measurements are generated from one set of true model parameters and perturbed with measurement errors $\varepsilon_C \sim \mathcal{N}(0, 0.005^2)$ and $\varepsilon_H \sim \mathcal{N}(0, 0.005^2)$. True values of the eight contaminant source parameters are listed in Table 4.

In this case, $f_H(\mathbf{m})$ is the numerical model built with MODFLOW and MT3DMS, while $f_L(\mathbf{m})$ is a data-driven surrogate built with a famous machine-learning method, that is, artificial neural network (ANN), using 2,000 sets of $f_H(\mathbf{m})$ simulation data (80% data are used for training, 10% are used for validation, and the rest 10% are used for testing). Here the Neural Net toolbox in MATLAB R2014a is used. The ANN has 30 hidden layers and is trained with the Levenberg–Marquardt algorithm (Hagan & Menhaj, 1994). By comparing $f_L(\mathbf{m})$ and $f_H(\mathbf{m})$ outputs at 200 prior samples, we can obtain $R^2 = 0.960$. Then we apply the $f_H(\mathbf{m})$ -MCMC, the $f_L(\mathbf{m})$ -MCMC, and the AMF-MCMC algorithms to infer the 28 unknown parameters, respectively. In both the $f_H(\mathbf{m})$ - and $f_L(\mathbf{m})$ -MCMC simulations, the number of parallel chains is set as 20, and the length of each chain is 12,000, which means 240,000 model evaluations in total for each simulation. In the AMF-MCMC simulation, the initial number of $f_H(\mathbf{m})$ evaluations is 100, the initial number of $f_L(\mathbf{m})$ evaluations is 200, the iteration number is 21, and at each iteration five sets of $f_H(\mathbf{m})$ data and five sets of $f_L(\mathbf{m})$ data are added to refine the GP system locally. In Figure 16, we plot the thinned chains of the $f_H(\mathbf{m})$ -MCMC simulation for the eight contaminant source parameters. It is clear that the Markov chains need about 100,000 $f_H(\mathbf{m})$ evaluations to converge to the true parameter values (black crosses). While the results of the $f_L(\mathbf{m})$ -MCMC simulation are significantly biased, although we have used 2,000 $f_H(\mathbf{m})$ evaluations to construct the low-fidelity model. With just 205 more $f_H(\mathbf{m})$ and 305 more $f_L(\mathbf{m})$ evaluations, a great improvement of the parameter estimation can be obtained by the AMF-MCMC algorithm.

4. Conclusions and Discussions

In this paper, we propose an efficient method for posterior exploration of hydrologic systems, that is, the AMF-MCMC algorithm. In the AMF-MCMC algorithm, data from both a high-fidelity model and a low-fidelity model are simultaneously fused to build a multifidelity GP system, based on which the MCMC simulation can be implemented quickly. As the region of interest is the posterior distribution, we successively add new parameter samples that are close to this region for the high- and low-fidelity model evaluations, which are then used to refine the multifidelity GP system locally. Finally, we can obtain an accurate estimation of the posterior distribution with a small number of the high-fidelity model evaluations.

To demonstrate the performance of the AMF-MCMC algorithm, we test three numerical cases in inverse modeling of hydrologic systems. Different types of low-fidelity models are used for illustration. In the first example, we estimate soil hydraulic and thermal parameters with the proposed method in a single ring infiltration experiment. Here the low-fidelity model is built with HYDRUS-2D with a coarser discretization. In the second example, we test a contaminant source identification problem that has multimodal posterior. Here the low-fidelity model is a data-driven surrogate. The third example handles a similar problem as the second one but considers more unknown parameters. In the three examples, the AMF-MCMC algorithm can obtain almost identical results as the $f_H(\mathbf{m})$ -MCMC algorithm but with a very low computational cost. Furthermore, the

adaptive multifidelity framework is universal in that it can be straightforwardly combined with other inverse or data assimilation methods, for example, different variants of Kalman filter/smooth (Chen & Oliver, 2012; Emerick & Reynolds, 2013; Evensen, 2007; Gu & Oliver, 2007).

When the number of unknown parameters is large (e.g., $N_m > 100$), many more training data from the high-fidelity model simulations are needed. Then the CPU time of model evaluations will be prohibitive. Moreover, the computational cost in GP system construction will also be huge as it scales cubically with the number of training data. Although we can apply some advanced GP methods, for example, sparse GP (Quinonero-Candela & Rasmussen, 2005; Snell & Ghahramani, 2006), to alleviate the computational cost of GP system construction, the AMF-MCMC algorithm proposed in this paper might still not be a good choice. In this situation, a more computationally appealing strategy is to adopt a method that assumes the posterior distribution to be multi-Gaussian (Elshall et al., 2014; Rasmussen et al., 2015) to roughly estimate the uncertainty. Such assumption is flawed when the posterior is strongly non-Gaussian or even multimodal, then people may take extra techniques like normal-score transform (Zhou et al., 2011) or local updating strategy (Zhang et al., 2018) for a remedy.

Acknowledgments

Computer codes and data used in this paper are available at: https://www.researchgate.net/publication/322714837_MATLAB_Codes_of_the_AMF-MCMC_Algorithm. This work is supported by the National Natural Science Foundation of China (grants 41771254 and 41571215). The authors would like to thank the Editor and anonymous reviewers for their constructive comments and suggestions, which significantly improve the quality of this work. The authors would also like to thank Andreas Klimke from University of Stuttgart for providing the sparse grid interpolation toolbox, Maziar Raissi from Brown University for providing the MATLAB codes of multifidelity GP, Jasper Vrugt from University of California, Irvine, for providing the MATLAB code of DREAM_(ZS), respectively.

References

Anderson, M. P., Woessner, W. W., & Hunt, R. J. (2015). *Applied groundwater modeling: Simulation of flow and advective transport*. San Diego, CA: Academic Press.

Asher, M. J., Croke, B. F. W., Jakeman, A. J., & Peeters, L. J. M. (2015). A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, 51, 5957–5973. <https://doi.org/10.1002/2015WR016967>

Brooks, S. P., & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4), 434–455. <https://doi.org/10.1080/10618600.1998.10474787>

Chen, Y., & Oliver, D. S. (2012). Ensemble randomized maximum likelihood method as an iterative ensemble smoother. *Mathematical Geosciences*, 44(1), 1–26. <https://doi.org/10.1007/s11004-011-9376-z>

Clark, M. P., Kavetski, D., & Fenicia, F. (2011). Pursuing the method of multiple working hypotheses for hydrological modeling. *Water Resources Research*, 47, W09301. <https://doi.org/10.1029/2010WR009827>

Coleman, T. F., & Li, Y. Y. (1996). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2), 418–445. <https://doi.org/10.1137/0806023>

Doherty, J., & Christensen, S. (2011). Use of paired simple and complex models to reduce predictive bias and quantify uncertainty. *Water Resources Research*, 47, W12534. <https://doi.org/10.1029/2011WR010763>

Efendiev, Y., Datta-Gupta, A., Ginting, V., Ma, X., & Mallick, B. (2005). An efficient two-stage Markov chain Monte Carlo method for dynamic data integration. *Water Resources Research*, 41, W12423. <https://doi.org/10.1029/2004WR003764>

Elshall, A. S., Pham, H. V., Tsai, F. T.-C., Yan, L., & Ye, M. (2014). Parallel inverse modeling and uncertainty quantification for computationally demanding groundwater-flow models using covariance matrix adaptation. *Journal of Hydrologic Engineering*, 20(8). [https://doi.org/10.1061/\(ASCE\)HE.1943-5584.0001126](https://doi.org/10.1061/(ASCE)HE.1943-5584.0001126)

Elshall, A. S., & Tsai, F. T.-C. (2014). Constructive epistemic modeling of groundwater flow with geological structure and boundary condition uncertainty under the Bayesian paradigm. *Journal of Hydrology*, 517, 105–119. <https://doi.org/10.1016/j.jhydrol.2014.05.027>

Elsheikh, A. H., Hoteit, I., & Wheeler, M. F. (2014). Efficient Bayesian inference of subsurface flow models using nested sampling and sparse polynomial chaos surrogates. *Computer Methods in Applied Mechanics and Engineering*, 269, 515–537. <https://doi.org/10.1016/j.cma.2013.11.001>

Emerick, A. A., & Reynolds, A. C. (2013). Ensemble smoother with multiple data assimilation. *Computers & Geosciences*, 55, 3–15. <https://doi.org/10.1016/j.cageo.2012.03.011>

Evensen, G. (2007). *Data assimilation: The ensemble Kalman filter*. Berlin, Germany: Springer.

Forrester, A. I. J., & Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1–3), 50–79. <https://doi.org/10.1016/j.paerosci.2008.11.001>

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4), 457–472. <https://doi.org/10.1214/ss/1177011136>

Gong, W., & Duan, Q. (2017). An adaptive surrogate modeling-based sampling strategy for parameter optimization and distribution estimation (ASMO-PODE). *Environmental Modelling & Software*, 95, 61–75. <https://doi.org/10.1016/j.envsoft.2017.05.005>

Horodetsky, A., & Marzouk, Y. (2016). Mercer kernels and integrated variance experimental design: Connections between Gaussian process regression and polynomial approximation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1), 796–828. <https://doi.org/10.1137/15M1017119>

Gu, Y., & Oliver, D. S. (2007). An iterative ensemble Kalman filter for multiphase fluid flow data assimilation. *SPE Journal*, 12(04), 438–446. <https://doi.org/10.2118/108438-PA>

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), 989–993. <https://doi.org/10.1109/72.329697>

Harbaugh, A. W., Banta, E. R., Hill, M. C., & McDonald, M. G. (2000). *MODFLOW-2000, the U.S. Geological Survey modular ground-water model user guide to modularization concepts and the ground-water flow process*. Reston, VA: U.S. Geological Survey. Retrieved from <https://pubs.usgs.gov/of/2000/0092/report.pdf>

Hu, S., Shi, L., Zha, Y., Williams, M., & Lin, L. (2017). Simultaneous state-parameter estimation supports the evaluation of data assimilation performance and measurement design for soil-water-atmosphere-plant system. *Journal of Hydrology*, 555, 812–831. <https://doi.org/10.1016/j.jhydrol.2017.10.061>

Ju, L., Zhang, J., Meng, L., Wu, L., & Zeng, L. (2018). An adaptive Gaussian process-based iterative ensemble smoother for data assimilation. *Advances in Water Resources*, 115, 125–135. <https://doi.org/10.1016/j.advwatres.2018.03.010>

Kang, P. K., Lee, J., Fu, X., Lee, S., Kitanidis, P. K., & Juanes, R. (2017). Improved characterization of heterogeneous permeability in saline aquifers from transient pressure data during freshwater injection. *Water Resources Research*, 53, 4444–4458. <https://doi.org/10.1002/2016WR020089>

Keating, E. H., Doherty, J., Vrugt, J. A., & Kang, Q. (2010). Optimization and uncertainty assessment of strongly nonlinear groundwater models with high parameter dimensionality. *Water Resources Research*, 46, W10517. <https://doi.org/10.1029/2009WR008584>

Kennedy, M. C., & O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1), 1–13. <https://doi.org/10.1093/biomet/87.1.1>

Klimke, A., & Wohlmuth, B. (2005). Algorithm 847: Spinterp: Piecewise multilinear hierarchical sparse grid interpolation in MATLAB. *ACM Transactions on Mathematical Software*, 31(4), 561–579. <https://doi.org/10.1145/1114268.1114275>

Krause, A., Singh, A., & Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9, 235–284.

Laloy, E., Rogiers, B., Vrugt, J. A., Mallants, D., & Jacques, D. (2013). Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov chain Monte Carlo simulation and polynomial chaos expansion. *Water Resources Research*, 49, 2664–2682. <https://doi.org/10.1002/wrcr.20226>

Laloy, E., & Vrugt, J. A. (2012). High-dimensional posterior exploration of hydrologic models using multiple-try DREAM (ZS) and high-performance computing. *Water Resources Research*, 48, W01526. <https://doi.org/10.1029/2011WR010608>

Le Gratiet, L., & Garnier, J. (2014). Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5), 365–386. <https://doi.org/10.1615/Int.J.UncertaintyQuantification.2014006914>

Linde, N., Ginsbourger, D., Irving, J., Nobile, F., & Doucet, A. (2017). On uncertainty quantification in hydrogeology and hydrogeophysics. *Advances in Water Resources*, 110, 166–181. <https://doi.org/10.1016/j.advwatres.2017.10.014>

Lu, D., Zhang, G. N., Webster, C., & Barbier, C. (2016). An improved multilevel Monte Carlo method for estimating probability distribution functions in stochastic oil reservoir simulations. *Water Resources Research*, 52, 9642–9660. <https://doi.org/10.1002/2016WR019475>

MacKay, D. J. (1992). Information-based objective functions for active data selection. *Neural Computation*, 4(4), 590–604. <https://doi.org/10.1162/neco.1992.4.4.590>

Mo, S., Lu, D., Shi, X., Zhang, G., Ye, M., Wu, J., & Wu, J. (2017). A Taylor expansion-based adaptive design strategy for global surrogate modeling with applications in groundwater modeling. *Water Resources Research*, 53, 10,802–10,823. <https://doi.org/10.1002/2017WR021622>

Mosleh, M., Rajagopal, R., & de Barros, F. P. J. (2015). Optimal allocation of computational resources in hydrogeological models under uncertainty. *Advances in Water Resources*, 83, 299–309. <https://doi.org/10.1016/j.advwatres.2015.06.014>

Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12, 513–522. <https://doi.org/10.1029/WR012i003p00513>

Nakhaei, M., & Šimůnek, J. (2014). Parameter estimation of soil hydraulic and thermal property functions for unsaturated porous media using the HYDRUS-2D code. *Journal of Hydrology and Hydromechanics*, 62(1), 7–15. <https://doi.org/10.2478/johh-2014-0008>

Narayan, A., Gittelson, C., & Xiu, D. (2014). A stochastic collocation algorithm with multifidelity models. *SIAM Journal on Scientific Computing*, 36(2), A495–A521. <https://doi.org/10.1137/130929461>

Palar, P. S., Tsuchiya, T., & Parks, G. T. (2016). Multi-fidelity non-intrusive polynomial chaos based on regression. *Computer Methods in Applied Mechanics and Engineering*, 305, 579–606. <https://doi.org/10.1016/j.cma.2016.03.022>

Parussini, L., Venturi, D., Perdikaris, P., & Karniadakis, G. E. (2017). Multi-fidelity Gaussian process regression for prediction of random fields. *Journal of Computational Physics*, 336, 36–50. <https://doi.org/10.1016/j.jcp.2017.01.047>

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N., & Karniadakis, G. E. (2017). Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical & Engineering Sciences*, 473(2198). <https://doi.org/10.1098/rspa.2016.0751>

Quinonero-Candela, J. Q., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6, 1939–1959.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Inferring solutions of differential equations using noisy multi-fidelity data. *Journal of Computational Physics*, 335, 736–746. <https://doi.org/10.1016/j.jcp.2017.01.060>

Rasmussen, J., Madsen, H., Jensen, K. H., & Refsgaard, J. C. (2015). Data assimilation in integrated hydrological modeling using ensemble Kalman filtering: Evaluating the effect of ensemble size and localization on filter performance. *Hydrology and Earth System Sciences*, 19(7), 2999–3013. <https://doi.org/10.5194/hess-19-2999-2015>

Razavi, S., Tolson, B. A., & Burn, D. H. (2012). Review of surrogate modeling in water resources. *Water Resources Research*, 48, W07401. <https://doi.org/10.1029/2011WR011527>

Refsgaard, J. C., Christensen, S., Sonnenborg, T. O., Seifert, D., Højberg, A. L., & Troldborg, L. (2012). Review of strategies for handling geological uncertainty in groundwater flow and transport modeling. *Advances in Water Resources*, 36, 36–50. <https://doi.org/10.1016/j.advwatres.2011.04.006>

Shi, X., Ye, M., Curtis, G. P., Miller, G. L., Meyer, P. D., Kohler, M., et al. (2014). Assessment of parametric uncertainty for groundwater reactive transport modeling. *Water Resources Research*, 50, 4416–4439. <https://doi.org/10.1002/2013WR013755>

Šimůnek, J., van Genuchten, M. T., & Šejna, M. (2008). Development and applications of the HYDRUS and STANMOD software packages and related codes. *Vadose Zone Journal*, 7(2), 587–600. <https://doi.org/10.2136/vzj2007.0077>

Smith, R. C. (2014). *Uncertainty quantification: Theory, implementation, and applications*. Philadelphia, PA: SIAM.

Smith, T. J., & Marshall, L. A. (2008). Bayesian methods in hydrologic modeling: A study of recent advancements in Markov chain Monte Carlo techniques. *Water Resources Research*, 44, W00805. <https://doi.org/10.1029/2007WR006705>

Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs, paper presented at Advances in Neural Information Processing Systems, Vancouver, Canada.

Sophocleous, M. (1979). Analysis of water and heat-flow in unsaturated-saturated porous media. *Water Resources Research*, 15, 1195–1206. <https://doi.org/10.1029/WR015i005p01195>

Stuart, A. M. (2010). Inverse problems: A Bayesian perspective. *Acta Numer*, 19, 451–559. <https://doi.org/10.1017/S0962492910000061>

Sun, A. Y., Wang, D., & Xu, X. (2014). Monthly streamflow forecasting using Gaussian process regression. *Journal of Hydrology*, 511, 72–81. <https://doi.org/10.1016/j.jhydrol.2014.01.023>

Van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal*, 44(5), 892–898. <https://doi.org/10.2136/sssaj1980.03615995004400050002x>

Vieux, B. E. (2001). Distributed hydrologic modeling using GIS. In *Water Science and Technology Library*, (Vol. 38, pp. 1–20). Dordrecht: Springer.

Vrugt, J. A. (2016). Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. *Environmental Modelling & Software*, 75, 273–316. <https://doi.org/10.1016/j.envsoft.2015.08.013>

Wagener, T., & Gupta, H. V. (2005). Model identification for hydrological forecasting under uncertainty. *Stochastic Environmental Research and Risk Assessment*, 19(6), 378–387. <https://doi.org/10.1007/s00477-005-0006-5>

Watson, T. A., Doherty, J. E., & Christensen, S. (2013). Parameter and predictive outcomes of model simplification. *Water Resources Research*, 49, 3952–3977. <https://doi.org/10.1002/wrcr.20145>

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Xu, T., & Valocchi, A. J. (2015). A Bayesian approach to improved calibration and prediction of groundwater models with structural error. *Water Resources Research*, 51, 9290–9311. <https://doi.org/10.1002/2015WR017912>

Xu, T., Valocchi, A. J., Ye, M., & Liang, F. (2017). Quantifying model structural error: Efficient Bayesian calibration of a regional groundwater flow model using surrogates and a data-driven error model. *Water Resources Research*, 53, 4084–4105. <https://doi.org/10.1002/2016WR019831>

Zeng, L., Shi, L., Zhang, D., & Wu, L. (2012). A sparse grid based Bayesian method for contaminant source identification. *Advances in Water Resources*, 37, 1–9. <https://doi.org/10.1016/j.advwatres.2011.09.011>

Zeng, X., Ye, M., Wu, J., Wang, D., & Zhu, X. (2018). Improved nested sampling and surrogate-enabled comparison with other marginal likelihood estimators. *Water Resources Research*, 54, 797–826. <https://doi.org/10.1002/2017WR020782>

Zeng, X. K., Ye, M., Burkhardt, J., Wu, J. C., Wang, D., & Zhu, X. B. (2016). Evaluating two sparse grid surrogates and two adaptation criteria for groundwater Bayesian uncertainty quantification. *Journal of Hydrology*, 535, 120–134. <https://doi.org/10.1016/j.jhydrol.2016.01.058>

Zha, Y., Yeh, T. C. J., Illman, W. A., Zeng, W., Zhang, Y., Sun, F., & Shi, L. (2018). A reduce-order successive linear estimator for geostatistical inversion and its application in hydraulic tomography. *Water Resources Research*, 54, 1616–1632. <https://doi.org/10.1002/2017WR021884>

Zhang, D., & Lu, Z. (2004). An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Loève and polynomial expansions. *Journal of Computational Physics*, 194(2), 773–794. <https://doi.org/10.1016/j.jcp.2003.09.015>

Zhang, J., Li, W., Lin, G., Zeng, L., & Wu, L. (2017). Efficient evaluation of small failure probability in high-dimensional groundwater contaminant transport modeling via a two-stage Monte Carlo method. *Water Resources Research*, 53, 1948–1962. <https://doi.org/10.1002/2016WR019518>

Zhang, J., Li, W., Zeng, L., & Wu, L. (2016). An adaptive Gaussian process-based method for efficient Bayesian experimental design in groundwater contaminant source identification problems. *Water Resources Research*, 52, 5971–5984. <https://doi.org/10.1002/2016WR018598>

Zhang, J., Lin, G., Li, W., Wu, L., & Zeng, L. (2018). An iterative local updating ensemble smoother for estimation and uncertainty assessment of hydrologic model parameters with multimodal distributions. *Water Resources Research*, 54, 1716–1733. <https://doi.org/10.1002/2017WR020906>

Zhang, J., Zeng, L., Chen, C., Chen, D., & Wu, L. (2015). Efficient Bayesian experimental design for contaminant source identification. *Water Resources Research*, 51, 576–598. <https://doi.org/10.1002/2014WR015740>

Zheng, C., & Wang, P. P. (1999). MT3DMS: A modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems; documentation and user's guide. DTIC Document. Retrieved from <http://www.geology.wisc.edu/courses/g727/mt3dmanual.pdf>

Zhou, H., Gomez-Hernandez, J. J., Franssen, H.-J. H., & Li, L. (2011). An approach to handling non-Gaussianity of parameters and state variables in ensemble Kalman filtering. *Advances in Water Resources*, 34(7), 844–864. <https://doi.org/10.1016/j.advwatres.2011.04.014>

Zhu, P., Shi, L., Zhu, Y., Zhang, Q., Huang, K., & Williams, M. (2017). Data assimilation of soil water flow via ensemble Kalman filter: Infusing soil moisture data at different scales. *Journal of Hydrology*, 555, 912–925. <https://doi.org/10.1016/j.jhydrol.2017.10.078>

Zhu, X., Narayan, A., & Xiu, D. (2014). Computational aspects of stochastic collocation with multifidelity models. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1), 444–463. <https://doi.org/10.1137/130949154>