Non-Gaussian parameter inference for hydrogeological models using Stein Variational Gradient Descent

M. Ramgraber[1,2], R. Weatherl[1,2], F. Blumensaat[3], M. Schirmer[1,2]

[1]Department of Water Resources and Drinking Water, Swiss Federal Institute of Aquatic Science and Technology (Eawag), Switzerland

[2]Centre for hydrogeology and geothermics (CHYN), University of Neuchâtel, Switzerland

[3]Department of Urban Water Management, Integrated Assessment and Modelling, Swiss Federal Institute of Aquatic Science and Technology (Eawag), Switzerland

Corresponding author: *Maximilian Ramgraber (max.ramgraber@eawag.ch).*

Key Points:

- This manuscript explores Stein Variational Gradient Descent (SVGD) for parameter inference in hydrogeological models
- We introduce an ensemble-based Jacobian approximation and test the algorithm in a synthetic and a real test case
- The algorithm performs well in high-dimensional and multi-modal, non-Gaussian settings

Abstract

The sustainable management of groundwater demands a faithful characterization of the subsurface. This, in turn, requires information which is generally not readily available. To bridge the gap between data need and availability, numerical models are often used to synthesize plausible scenarios not only from direct information but also additional, indirect data. Unfortunately, the resulting system characterizations will rarely be unique. This poses a challenge for practical parameter inference: Computational limitations often force modelers to resort to methods based on questionable assumptions of Gaussianity, which do not reproduce important facets of ambiguity such as Pareto fronts or multi-modality. In search of a remedy, an alternative could be found in *Stein Variational Gradient Descent*, a recent development in the field of statistics. This ensemble-based method iteratively transforms a set of arbitrary particles into samples of a potentially non-Gaussian posterior, provided the latter is sufficiently smooth. A prerequisite for this method is knowledge of the Jacobian, which is usually exceptionally expensive to evaluate. To address this issue, we propose an ensemble-based, localized approximation of the Jacobian. We demonstrate the performance of the resulting algorithm in two cases: a simple, bimodal synthetic scenario, and a complex numerical model based on a real-world, pre-alpine catchment. Promising results in both cases – even when the ensemble size is smaller than the number of parameters – suggest that *Stein Variational Gradient Descent* can be a valuable addition to hydrogeological parameter inference.

1 Introduction

Parameter estimation for numerical models can synthesize different types of information into a physically plausible narrative. This is of particular relevance for the discipline of hydrogeology, where informed management demands detailed knowledge of the system, but direct measurements of the relevant subsurface properties are scarce and often of limited spatial representativeness (e.g., Rubin, 2003). The process of inferring subsurface properties from dependent information such as hydraulic head, chemical concentrations, or flow is known as inverse modelling (e.g., Carrera et al., 2005).

Unfortunately, as a consequence of the exceptional complexity of many hydrogeological systems (Figure 1), there usually exists more than a single plausible explanation for the observed data (Linde et al., 2015, 2017; Moeck et al., 2020). Variations in aquifer depth, sediment properties, atmospheric and hydrogeological forcing, anthropogenic influences, and complex geological features interact with each other and can create similar hydraulic responses in different arrangements. The consequence of this has been summarized succinctly by Poeter & Townsend (1994): "*A true evaluation of the possible subsurface configurations and their impact on the decision at hand is the only honest approach to groundwater analyses.*" and hence surmised that "*The era of drawing conclusions on the basis of deterministic flow and transport models has come to a close*".

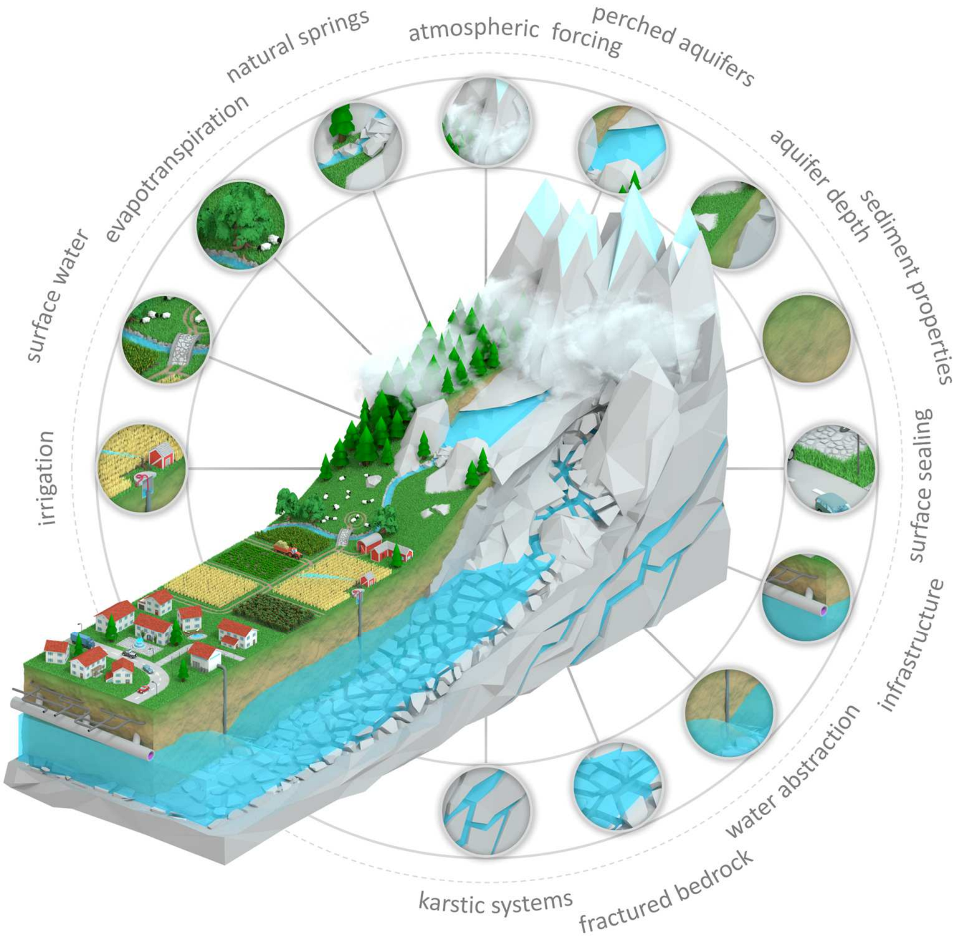


Figure 1. Complex and interacting aspects in a mountainous hydrogeological system. When the presence, properties, and extent of these aspects is not sufficiently quantified, they become sources of uncertainty for hydrogeological models.

Where deterministic models only seek a single promising model configuration, stochastic approaches based on Bayesian statistics explore multiple alternative configurations at once. This process hopes to identify ambiguities in order to endow model predictions with reliable uncertainty estimates. Unfortunately, 25 years later, Poeter & Townsend (1994)'s prediction has yet to fully come to pass. While the need for probabilistic groundwater models has been widely acknowledged (e.g., Cirpka & Valocchi, 2016; Renard, 2007; Sanchez-Vila & Fernàndez-Garcia, 2016), the complexity of representing the hydrogeological system – and, by consequence, its uncertainties – remains an obstacle for the wide-scale adoption of Bayesian methods.

In Bayesian statistics, the plausibility of different narratives – as defined by model parameterizations – is represented through *probability density functions* (pdf). Bayes' theorem formalizes the synthesis of a so-called *posterior* from initial belief (the *prior*) and new data (the *likelihood*). Since it has no analytical solution in the general case, its practical use often demands approximations and simplifications. Among the most elegant is *Gaussianity*, which permits an analytical solution provided that the numerical model is linear, and that all pdfs involved are Gaussian. This assumption underlies the popular Ensemble Kalman Filter (EnKF: Evensen, 1994, 2003), which has proven easy to implement and highly robust to small ensemble sizes. As a consequence, it quickly gained popularity in the hydrogeological community (e.g., Gu & Oliver, 2007; Hendricks Franssen et al., 2011; Keller et al., 2018; Reichle et al., 2002). Unfortunately, the assumption of Gaussianity implies both unimodality (*there exists a single most probable solution*) and full support (*no solution is impossible*). Both assumptions are potentially problematic: the former because it cannot adequately represent the existence of distinct, equivalent solutions in the form of Pareto fronts or separate probability modes; the latter for parameters with strict physical limits.

It may seem expedient, then, to turn our attention to more general approaches such as *Markov Chain Monte Carlo* (MCMC: e.g., Foreman-Mackey et al., 2013; Smith & Marshall, 2008) or *particle filters* (PF: e.g., van Leeuwen, 2009; van Leeuwen et al., 2019). These methods can theoretically approximate arbitrary distributions but suffer from practical limitations of their own. Fundamentally, both methods suffer in systems with high-dimensionality, although the specific symptoms vary: MCMC methods often display large autocorrelations if the proposal distributions are not sufficiently well-tuned, which reduces the sample generation efficiency significantly. Possible remedies are found in Hamiltonian Monte Carlo (e.g., Betancourt, 2018), which exploit Jacobian information, or approaches like the affine-invariant ensemble sampler for MCMC *emcee* (Foreman-Mackey et al., 2013), which can restrict itself to a limited subspace. The ensembles of PFs, on the other hand, tend to quickly degenerate and collapse in high-dimensional systems (e.g., Arulampalam et al., 2002; Bengtsson et al., 2008), and may require pragmatic solutions which threaten to corrupt the inference (Moradkhani et al., 2005; Ramgraber et al., 2019; Vrugt et al., 2013). As such, these computational limitations render both methods less efficient in systems with limited computational resources than comparable Gaussian-based approaches.

In search of a free lunch, we would desire an inference algorithm which combines the strengths of the above: the efficiency and robustness of the EnKF in face of small ensemble sizes, and the PF's/MCMC's ability to explore non-Gaussian distributions. Stein Variational Gradient Descent (Liu & Wang, 2016), a relatively recent development in the computational sciences, may be an interesting step in this direction. Based on Kernelized Stein Discrepancy (Chwialkowski et al., 2016; Liu et al., 2016), it yields a surprisingly simple gradient descent algorithm capable of iteratively transforming an arbitrary ensemble of particles into samples of the posterior. With a few small adjustments, we shall see that it can share the EnKF's ability to scale the complexity of the inference problem by restricting the analysis to a parameter-subspace whose dimensionality depends on the number of available particles, while at the same time being able to approximate non-Gaussian distributions. In the following, we will re-derive the algorithm, then propose adaptations and approximations required to render it tractable in practice. Afterwards, we will demonstrate its performance in a simple, bi-modal synthetic scenario, as well as in a highly complex pre-alpine catchment. Finally, we will discuss the results and provide an outlook for future research. First, however, we will present the nomenclature used in this study.

2 Theory

2.1 Nomenclature

In this study, we will use bold font to denote vectors or matrices and will refer to column vectors ($\boldsymbol{\theta} = [\theta_1, \ldots, \theta_d]^\top$) unless otherwise specified. The symbol $\boldsymbol{\theta}$ denotes the vector of model parameters, and the variable $\boldsymbol{x}$ denotes model states. Data or observations are represented by $\boldsymbol{y}$. Standard font (e.g., θ) refers to scalar-valued variables. For functions, we shall refer to the function as an object by $\boldsymbol{f}$, and to its output by $\boldsymbol{f}(\boldsymbol{\theta})$. Functions with multiple arguments (e.g., $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$), for which one argument is assumed fixed, are denoted by a dot in its arguments (e.g., $k(\cdot, \boldsymbol{\theta}')$ for fixed $\boldsymbol{\theta}'$). $\|\boldsymbol{\theta}\|$ refers to the norm and $|\boldsymbol{\theta}|$ to the absolute value of $\boldsymbol{\theta}$. Superscripts in parentheses $\boldsymbol{\theta}^{(d)}$ refer to the d-th entry of $\boldsymbol{\theta}$. Capitalized roman normal symbols refer to integer variables: D to the dimensionality of parameter space (number of model parameters), O to the dimensionality of observation space (number of state observations), and N to the number of particles (ensemble size).

2.2 Stein Variational Gradient Descent

In the following, we will present the Stein Variational Gradient Descent (SVGD) algorithm following the derivations outlined in Liu et al. (2016) and Liu & Wang (2016). In short, SVGD iteratively transforms samples of an arbitrary reference distribution into samples of the posterior. This process may bear superficial similarity to filter techniques, but is based on a crucial difference: instead of sequentially adding information through re-weighting steps (think *treasure map*: specifying the steps to the target one by one), it 'homes in' on the posterior distribution iteratively (think *navigation system*: constantly reorienting itself towards the target).

The algorithm is based on an incremental particle flow which iteratively transforms an ensemble of initial samples into posterior samples:

$$\boldsymbol{\theta}_i = \boldsymbol{T}(\boldsymbol{\theta}_{i-1}) = \boldsymbol{\theta}_{i-1} + \varepsilon \boldsymbol{\phi}^*(\boldsymbol{\theta}_{i-1}) \tag{1}$$

where the subscript i denotes the current iteration number, ε is a small scalar increment, and $\boldsymbol{\phi}^*: \mathbb{R}^D \to \mathbb{R}^D$ is a vector field whose pointwise evaluations $\boldsymbol{\phi}^*(\boldsymbol{\theta}_{i-1})$ designate the flow direction for each particle.

This vector field $\boldsymbol{\phi}^*$ is the key ingredient of SVGD. As we shall see in the following, it can be found through a function optimization on the space of vector fields/infinitesimal transformations. We identify the infinitesimal transformation that maximally reduces the Kullback-Leibler divergence (KLD) to the target posterior. The associated vector field thus corresponds to the negative functional gradient of the KLD, and its norm defines a discrepancy measure called the Kernelized Stein Discrepancy (KSD). The resulting equation for $\boldsymbol{\phi}^*$ is surprisingly simple, providing the particle flow directions for an infinitesimally small step towards the posterior distribution. However, in order to understand the derivation of the algorithm, we must introduce the concept of a *Reproducing Kernel Hilbert Space* (*RKHS*).

2.2.1 Reproducing Kernel Hilbert Spaces

RKHS are special, infinite-dimensional function spaces with several properties which make them interesting for functional optimization tasks – tasks, in which we want to find functions which fulfil certain requirements. There are several different ways to define a RKHS $\mathcal{H}$. In this study, we adopt the definition used in Liu et al. (2016). This definition is based on the spectral decomposition of a positive definite, symmetric kernel $k(\boldsymbol{\theta}, \boldsymbol{\theta}'): \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. An example of such a kernel is the *radial basis function* (RBF) kernel:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp\left(-\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2}{2h^2}\right) \tag{2}$$

where h^2 is the kernel's bandwidth. Kernels can be regarded as similarity metrics between two particles $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$: if the particles are identical, the kernel yields 1, and the more different they are, the closer the kernel's output will be to zero. According to *Mercer's theorem*, any symmetric, positive semi-definite kernel is associated with an inner product on some Hilbert space $\mathcal{H}$, obtained through spectral decomposition of the Hilbert-Schmidt integral operator (e.g., Schölkopf & Smola, 2001; Werner, 2018):

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{l=1}^{\infty} \lambda_l e_l(\boldsymbol{\theta}) e_l(\boldsymbol{\theta}') \tag{3}$$

This expresses the kernel as an infinite series of orthonormal eigenfunctions e_l and eigenvalues λ_l. These eigenfunctions can be interpreted as an orthonormal basis spanning up an infinite-dimensional RKHS $\mathcal{H}$ which comprises of linear combinations of its eigenfunctions $f(\boldsymbol{\theta}) = \sum_{l=1}^{\infty} f_l e_l(\boldsymbol{\theta})$ with $\sum_{l=1}^{\infty} {f_l}^2/\lambda_l < \infty$ and an inner product $\langle f, g \rangle_{\mathcal{H}} = \sum_{l=1}^{\infty} f_l g_l/\lambda_l$ between $f(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta}) = \sum_{l=1}^{\infty} g_l e_l(\boldsymbol{\theta})$. This also defines a norm $\|f\|_{\mathcal{H}}$ where $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}} = \sum_{l=1}^{\infty} {f_l}^2/\lambda_l$.

Equation 3 may then be interpreted as an inner product between two vectors $\boldsymbol{k}(\cdot, \boldsymbol{\theta})$ and $\boldsymbol{k}(\cdot, \boldsymbol{\theta}')$ in $\mathcal{H}$. Since their embedding space $\mathcal{H}$ is infinite-dimensional, these vectors will have infinitely many entries:

$$\boldsymbol{k}(\boldsymbol{\theta}, \cdot) = \left[\sqrt{\lambda_1} e_1(\boldsymbol{\theta}), \ldots, \sqrt{\lambda_\infty} e_\infty(\boldsymbol{\theta})\right]^\top \tag{4}$$

$$\boldsymbol{k}(\cdot,\boldsymbol{\theta}') = \left[\sqrt{\lambda_1}e_1(\boldsymbol{\theta}'),\dots,\sqrt{\lambda_\infty}e_\infty(\boldsymbol{\theta}')\right]^\top \tag{5}$$

Why is this useful? In machine learning literature, particularly for classification tasks (e.g., *support vector machines*: Schölkopf & Smola, 2001), it is common to extract *features* (here: $\sqrt{\lambda_l}e_l(\boldsymbol{\theta})$) from an *input data set* (here: $\boldsymbol{\theta}$). The larger the amount of independent, extracted features, the easier the classification becomes. In a RKHS, the number of these features is infinite. And if the only operation on these features required is an inner product, we need not even compute them – an evaluation of the kernel would yield the desired result. We can verify that an inner product between Equation 4 and Equation 5 yields Equation 3, and retrieve one of the fundamental properties of a RKHS $\mathcal{H}$:

$$k(\boldsymbol{\theta},\boldsymbol{\theta}') = \langle \boldsymbol{k}(\boldsymbol{\theta},\cdot),\boldsymbol{k}(\cdot,\boldsymbol{\theta}')\rangle_{\mathcal{H}} = \sum_{l=1}^{\infty}\sqrt{\lambda_l}e_l(\boldsymbol{\theta})\sqrt{\lambda_l}e_l(\boldsymbol{\theta}') = \sum_{l=1}^{\infty}\lambda_l e_l(\boldsymbol{\theta})e_l(\boldsymbol{\theta}') \tag{6}$$

For the purpose of functional optimization, we are interested in the functions defined in the RKHS. $\mathcal{H}$ contains scalar-valued functions f mapping from the parameter space $(f:\mathbb{R}^D \to \mathbb{R})$ which are constructed through linear combinations of its basis, the eigenfunctions:

$$f(\boldsymbol{\theta}) = \sum_{l=1}^{\infty} f_l e_l(\boldsymbol{\theta}) \tag{7}$$

where f_l are arbitrary real scalars. These functions are uniquely defined by a vector $\boldsymbol{f}(\cdot)$ in $\mathcal{H}$

$$\boldsymbol{f}(\cdot) = \left[f_1/\sqrt{\lambda_1},\dots,f_\infty/\sqrt{\lambda_\infty}\right]^T \tag{8}$$

and can be retrieved by taking an inner product with Equation 5 (replacing $\boldsymbol{\theta}'$ with $\boldsymbol{\theta}$). This defines the RKHS's eponymous *reproducing property*:

$$f(\boldsymbol{\theta}) = \langle \boldsymbol{f}(\cdot),\boldsymbol{k}(\cdot,\boldsymbol{\theta})\rangle_{\mathcal{H}} = \sum_{l=1}^{\infty}\frac{f_l}{\sqrt{\lambda_l}}\sqrt{\lambda_l}e_l(\boldsymbol{\theta}) = \sum_{l=1}^{\infty} f_l e_l(\boldsymbol{\theta}) \tag{9}$$

With the fundamentals of RKHS defined, let us proceed to the derivation of the algorithm.

2.2.2 Deriving the algorithm

SVGD is derived from a metric called *Kernelized Stein Discrepancy* (*KSD*: Chwialkowski et al., 2016; Liu et al., 2016) $\mathbb{S}(q||p)$ between two probability distributions q and p. This metric yields a measure of discrepancy between the two distributions, provided that we have an ensemble of samples from q and are able to evaluate the gradient of the logarithm of p at least pointwise. In our application, q will always be some intermediate distribution from which we assume our samples are drawn, and p will be the target posterior.

$$\mathbb{S}(q||p) = \max_{\boldsymbol{\phi}\in\mathcal{F}} \left\{ \left[\mathbb{E}_{\boldsymbol{\theta}\sim q} \left[\operatorname{trace} \boldsymbol{A}_p \boldsymbol{\phi}(\boldsymbol{\theta}) \right] \right]^2 \right\} \tag{10}$$

In Equation 10, $\mathbb{E}_{\boldsymbol{\theta}\sim q}$ refers to the expectation under the assumption that the particles $\boldsymbol{\theta}$ are sampled from q, $\boldsymbol{\phi}$ is a vector field on parameter space, representing an infinitesimal transformation, and $\boldsymbol{A}_p$ is a linear operator:

$$\boldsymbol{A}_p \boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{\phi}(\boldsymbol{\theta}) [\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta})]^\top + \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{\phi}(\boldsymbol{\theta}) \tag{11}$$

where $\boldsymbol{\nabla}_{\boldsymbol{\theta}} = \left[\partial/\partial\boldsymbol{\theta}^{(1)}, \dots, \partial/\partial\boldsymbol{\theta}^{(1)}\right]^\top$ denotes the partial derivative operator evaluated at $\boldsymbol{\theta}$. We have provided a detailed derivation of Equation 10 in Appendix 1 (Supporting Information). The challenging part in Equation 10 is the functional optimization, specifically the need to find the vector field $\boldsymbol{\phi}^*$ which maximizes the violation of Stein's identity. Fortunately, this is where the properties of the RKHS prove advantageous. If we assume the family of functions $\mathcal{F}$ are the functions we can define in a RKHS (Equation 8 and 9), the functional optimization in Equation 10 has a closed-form solution:

$$\boldsymbol{\phi}^*(\boldsymbol{\theta}') = \mathbb{E}_{\boldsymbol{\theta}\sim q}[k(\boldsymbol{\theta}, \boldsymbol{\theta}') \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \boldsymbol{\nabla}_{\boldsymbol{\theta}} k(\boldsymbol{\theta}, \boldsymbol{\theta}')] \tag{12}$$

We have re-derived this solution in detail in Appendix 2 (Supporting Information). The vector-valued function $\boldsymbol{\phi}^*$ defines a vector field over the parameter space $\mathbb{R}^D$, and assigns to each position a D-dimensional vector or direction which maximizes the violation of Stein's identity.

SVGD exploits this information to implement a particle flow which gradually transforms the distribution q into the distribution p, the posterior. It can be shown (Liu & Wang, 2016), that for linear invertible transformations the directions $\boldsymbol{\phi}^*(\boldsymbol{\theta}')$ of the vector field $\boldsymbol{\phi}^*$ correspond to the steepest descent directions of the *Kullback-Leibler divergence* (KLD). We have re-derived this for the reader's convenience in Appendix 3 (Supporting Information). Using the transformation in Equation 1, we establish an iterative particle flow through parameter space. The steepest descent directions $\boldsymbol{\phi}^*(\boldsymbol{\theta})$ are obtained by taking an ensemble approximation of Equation 12:

$$\boldsymbol{\phi}^*(\boldsymbol{\theta}) = \frac{1}{N} \sum_{j=1}^{N} k(\boldsymbol{\theta}_j, \boldsymbol{\theta}) \boldsymbol{\nabla}_{\boldsymbol{\theta}_j} \log p(\boldsymbol{\theta}_j) + \boldsymbol{\nabla}_{\boldsymbol{\theta}_j} k(\boldsymbol{\theta}_j, \boldsymbol{\theta}) \tag{13}$$

The only expensive variable to evaluate is the gradient of the logposterior at the particle positions $\boldsymbol{\nabla}_{\boldsymbol{\theta}_j} \log p(\boldsymbol{\theta}_j)$. In general cases, where no analytic form for the logposterior or its derivative are available, we must resort to approximations of this gradient. We will investigate a few approaches towards this end in the following section.

3 Algorithmic approximations

3.1 Posterior gradient $\boldsymbol{\nabla_\theta} \log p(\boldsymbol{\theta})$

While it is in principle possible to approximate $\boldsymbol{\nabla_\theta} \log p(\boldsymbol{\theta})$ directly from logposterior estimates, it may not always be advantageous to do so. If the logprior is differentiable, we can limit the approximation to the gradient of the loglikelihood or even just the Jacobian matrix, thus avoiding unnecessary approximation error. Towards this end, we can reformulate Bayes' Theorem to calculate the logposterior gradient as

$$\boldsymbol{\nabla_\theta} \log f(\boldsymbol{\theta}|\boldsymbol{y}) = \boldsymbol{\nabla_\theta} \log f(\boldsymbol{\theta}) + \boldsymbol{\nabla_\theta} \log f(\boldsymbol{y}|\boldsymbol{\theta}) \tag{14}$$

where $f(\boldsymbol{\theta}|\boldsymbol{y}) \coloneqq p(\boldsymbol{\theta})$ is the posterior pdf, $f(\boldsymbol{\theta})$ the prior pdf, and $f(\boldsymbol{y}|\boldsymbol{\theta})$ the likelihood. Since the logprior gradient is often available in closed form, we are left with finding the loglikelihood gradient. If we assume multivariate Gaussian likelihoods, we have:

$$f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^O \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{y}_{sim})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{y}_{sim})\right) \tag{15}$$

where O is the number of observations (the dimensionality of observation space), $\boldsymbol{\Sigma}^{-1}$ is the inverse of the error covariance matrix, and $\boldsymbol{y}$ and $\boldsymbol{y}_{sim}$ refer to the observed and simulated states. If we first take the logarithm and then the partial derivatives, we obtain:

$$\boldsymbol{\nabla_\theta} \log f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{2} \boldsymbol{\nabla_x} \boldsymbol{y}_{sim}{}^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{y}_{sim}) \tag{16}$$

If we simulate the states with a numerical model which takes as input parameters $\boldsymbol{x}$, i.e.:

$$\boldsymbol{y}_{sim} = \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}) \tag{17}$$

where we simplified notation slightly by implying that the model simulates the observed states directly. In practice, the model would generate the full state space (i.e., a time series of water table fields), and we would extract only the relevant dimensions/entries – for example at the locations of observation wells at certain times. Plugging this into Equation 16 and defining the Jacobian $\boldsymbol{J}(\boldsymbol{\theta}) = \boldsymbol{\nabla_\theta} \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})^\top$, we obtain:

$$\boldsymbol{\nabla_\theta} \log f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{2} \boldsymbol{J}(\boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{y}_{obs} - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})\right) \tag{18}$$

As such, we can obtain the logposterior gradient with local approximations of the Jacobian matrix.

3.2 Jacobian matrix $\nabla_{\boldsymbol{\theta}}\boldsymbol{\mathcal{M}}(\boldsymbol{\theta})$

The computational bottleneck for the solution of Equation 18, and by extension Equation 13, is the Jacobian $\boldsymbol{J}(\boldsymbol{\theta})$, an $O \times D$ matrix, which is not generally available in closed form. Some recent developments like automatic differentiation (e.g., Margossian, 2019) hold promise for future applications, but are model-intrusive and not yet widely supported.

Instead, we can explore non-intrusive approximations of the Jacobian. The standard numerical approach consists of perturbing the parameter vector $\boldsymbol{\theta}$ by a small increment along each dimension, then filling the Jacobian matrix with the resulting two-point (or three-point) finite difference derivatives (e.g., Wendt et al., 2009). While this numerical differentiation can yield very precise approximations, it quickly becomes computationally unfeasible for models with many parameters: To obtain the set of local Jacobians, we would have to run the model $N(D+1)$ times (or $N(2D+1)$ times for three-point derivatives) in each iteration. For complex, computationally demanding models, we generally cannot afford more than N model evaluations.

As such, we may wish to estimate the Jacobian directly from the ensemble, using only the N model evaluations $\boldsymbol{\mathcal{M}}(\boldsymbol{\theta})$. One such approach has been used by Chen & Oliver (2013) and White (2018), approximating the Jacobian based on prior and model error information, and each ensemble member's deviation from the mean. This approach can be useful in many applications, but is unfortunately based on the assumption of Gaussianity, and thus squanders the non-Gaussian properties which motivated our exploration of SVGD in the first place. Pulido et al. (2019) suggest an alternative approach which defines the observation operator (analogous to our model $\boldsymbol{\mathcal{M}}$) in a RKHS, then shifts the derivative operator to the kernel:

$$\boldsymbol{J}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n) \nabla_{\boldsymbol{\theta}} k(\boldsymbol{\theta}, \boldsymbol{\theta}_n) \tag{19}$$

This approach can also be interpreted as the derivative of a radial basis function approximation with vector-valued coefficients $\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)$. A similar expression can also be obtained by replacing the $\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)$ with a vector of coefficients determined to ensure the RBF interpolation reproduces the model output surface exactly at the particles. This approach can be very useful, but has a few potential caveats:

First and foremost, RBF approximations taper off towards zero when moving away from the ensemble ($\lim_{\boldsymbol{\theta}\to\infty} k(\boldsymbol{\theta}, \boldsymbol{\theta}_n) = 0$), which is undesirable for variables with non-zero limits. This can be addressed by pre-treating the data with a deterministic, differentiable routine such as multilinear regression, then interpolating only the residuals. A second issue is that a particle's local derivatives are informed exclusively by its neighbors, since the kernel derivative evaluated at its own center is zero ($\boldsymbol{\nabla}_{\boldsymbol{\theta}} k(\boldsymbol{\theta}, \boldsymbol{\theta}) = 0$). This can be problematic for remote or isolated particles. Similarly, the indirect nature of an RBF interpolation's derivatives does not exploit gradient information between the particles. The result depends critically on the chosen bandwidth (e.g., Mongillo, 2011) which can render the approach less robust than desired.

Consequently, we propose a different ensemble-based approximation, endeavoring to retain the localization of Pulido et al. (2019)'s approach while exploiting relative differences between the particles:

$$\tilde{\boldsymbol{J}}(\boldsymbol{\theta}_n) = \frac{P}{N} \sum_{m=1}^{N} \frac{\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)}{\|\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\|} \cdot \frac{\|\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\|}{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|} \cdot \frac{{\boldsymbol{\theta}_m}^{\top} - {\boldsymbol{\theta}_n}^{\top}}{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|} \tag{20}$$

where P is the expected rank of the Jacobian (usually either the D or $N-1$, whichever is smaller). The sum's first fraction is the normalized vector from particle $\boldsymbol{\theta}_n$ to particle $\boldsymbol{\theta}_m$ in observation space, the second fraction the scalar gradient between observation- and parameter space, and the third fraction the normalized vector in parameter space. We can simplify this to:

$$\tilde{\boldsymbol{J}}(\boldsymbol{\theta}_n) = \frac{P}{N} \sum_{m=1}^{N} \frac{\left(\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\right) \cdot (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^{\top}}{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|^2} \tag{21}$$

The factor P/N is composed of the arithmetic average's normalization constant ($1/N$) and a correction factor for the fact that each vector contributes at most one rank to the Jacobian (P). In linear systems, equation above should converge against the correct Jacobian for $N \to \infty$ and an isotropic particle arrangement. In nonlinear systems, we further need the assumption that $\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|$ is infinitesimally small, or a localization term which restricts the contributions of far-away particles ($w_m \to 0 \; for \; \|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\| \to \infty$):

$$\tilde{\boldsymbol{J}}(\boldsymbol{\theta}_n) = P \sum_{m=1}^{N} w_m \frac{\left(\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\right) \cdot (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^{\top}}{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|^2} \tag{22}$$

$$w_m = k_{\boldsymbol{\theta}_n}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_m) \Big/ \sum\nolimits_{l \neq n} k_{\boldsymbol{\theta}_n}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_l) \tag{23}$$

where w_m is some normalized, distance-based weight, for example obtained through a kernel $k_{\boldsymbol{\theta}_n}$. In practice, these assumptions will not generally be met. Possible consequences are that the Jacobian matrix may be biased if the parameter space vectors are not directionally isotropic, and the magnitude of the derivatives may be erroneous if the system is nonlinear and the particles are spaced too far apart. However, comparing this approach to a RBF interpolation, we found that it performed more robustly with regards to different bandwidth sizes and non-smooth conditions for the synthetic test case presented in Section 4. A small code example comparing both approaches is provided in Supporting Information (Appendix S6). Pseudo-code for the Jacobian approximation is provided in Figure 2.

Step 1: Create empty Jacobian
$J(\boldsymbol{\theta}_n) = zeros(O \times D)$

For particle m from 1 to N, if $m \neq n$:

Step 2: Create difference vectors
$\boldsymbol{u}_m = \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)$
$\boldsymbol{v}_m = \boldsymbol{\theta}_m - \boldsymbol{\theta}_n$

Step 3: Create their normalized variants
$\tilde{\boldsymbol{u}}_m = \boldsymbol{u}_m / \|\boldsymbol{u}_m\|$
$\tilde{\boldsymbol{v}}_m = \boldsymbol{v}_m / \|\boldsymbol{v}_m\|$

Step 4: Calculate the scalar gradient
$g_m = \|\boldsymbol{u}_m\| / \|\boldsymbol{v}_m\|$

Step 5: Determine gradient matrix
$\tilde{\boldsymbol{J}}_m = \tilde{\boldsymbol{u}}_m g_m \tilde{\boldsymbol{v}}_m{}^\top$

Step 6: *Find individual kernel bandwidth*
Set h of $k_{\boldsymbol{\theta}_n}$ to k-th median distance to other particles

Step 7: Calculate kernel weight
$w = k_{\boldsymbol{\theta}_n}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_m) / \sum_{l \neq n} k_{\boldsymbol{\theta}_n}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_l)$ (see Equation 23)

Step 8: Add contribution to Jacobian
$\boldsymbol{J}(\boldsymbol{\theta}_n) = \boldsymbol{J}(\boldsymbol{\theta}_n) + w\tilde{\boldsymbol{J}}_m$

Figure 2. Pseudo-code for the Jacobian approximation used in this study. Without additional model runs, evaluations of the Jacobian are only possible at the particle positions.

3.3 Gradient Descent algorithm

For an efficient inference with SVGD, we do not only require the descent directions $\boldsymbol{\phi}^*(\boldsymbol{\theta})$ (Equation 13), but also an adaptive scheme to adjust the step-size ε. If the step-size is too small, the algorithm may require too many iterations to be useful. If the step-size is too large, the algorithm may overshoot, start oscillating, and fail to locate a high-probability region at all. As such, we would like to adjust the step-size dynamically.

Many such algorithms exist. Methods like *adaptive moment estimation* (ADAM: Kingma & Ba, 2015) or *adaptive subgradient methods* (AdaGrad: Duchi et al., 2011) have proven successful for optimization in machine learning algorithms, being capable of dynamically adjusting the gradient descent to improve efficiency. Unfortunately, they often employ individual step-sizes for each parameter space dimension or otherwise alter the gradient vectors at each position through momentum. Since the theory derived above assumes a scalar, uniform ε at each iteration, we construct an alternative descent algorithm for this study which abides by these restrictions:

$$a_{i,n} = \alpha^{\left(\langle\frac{\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})}{\|\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})\|},\frac{\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})}{\|\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})\|}\rangle-\beta\right)} \min\left(1,\frac{\|\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})\|}{\|\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})\|}\right) \tag{24}$$

$$\varepsilon_i = \varepsilon_{i-1}\frac{1}{N}\sum_{n=1}^{N} a_{i,n} \tag{25}$$

This step-size update algorithm does not affect the gradient direction but may require some explanation to become intuitive. It requires two hyperparameters: an acceleration rate $\alpha > 1$, and a similarity cutoff $0 \leq \beta < 1$. At each iteration, the previous step-size ε_{i-1} is rescaled by a factor (Equation 25) corresponding to the ensemble mean of all acceleration proposals $a_{i,n}$ (Equation 24). These acceleration proposals are composed of two terms: the first term compares the directions of two subsequent descent vectors, proposing acceleration if the directions are sufficiently similar and deceleration if they are not; the second term compares the norm of two subsequent descent vectors, proposing deceleration if the norm (and thus velocity) of the vector increases.

For the first part, we exponentiate α by the inner product between the normalized current descent direction $\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})/\|\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})\|$ and the normalized previous descent direction $\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})/\|\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})\|$. This compares the similarity of both vectors and accelerates or slows the descent accordingly. Since a naïve inner product would only stop accelerating for turns sharper than 90° – and we may want to stop accelerating long before that – the second hyperparameter β is subtracted from the inner product. A cutoff of $\beta = 0.75$, for example, restricts acceleration to a cone of about 40° around the previous vector. For the second part, if the norm of the descent algorithm is increasing ($\|\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})\| < \|\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})\|$), the step-size should be reduced proportionally to reduce the risk of shooting past the optimum if the descent direction remains the same.

3.4 Pseudocode

To summarize the algorithmic approximations used in this study, pseudo-code for the algorithm is provided in Figure 3.

Initialization

- Draw N particles from the prior $f(\boldsymbol{\theta})$
- Generate (or draw) the initial states $\boldsymbol{x}_0$
- Define initial stepsize ε_0, acceleration rate α, and cutoff β
- Set iteration counter $i = 0$ and iteration Boolean $iterate = True$

While $iterate = True$**:**

$i \rightarrow i + 1$

Step 1: Numerical simulation

For particle n **from** 1 **to** N:

1. Use pre-processors to calculate auxiliary variables (if applicable)
2. Simulate and extract observed variables $\boldsymbol{x}_n = \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n, \boldsymbol{x}_0, \dots)$

Step 2: Determine Gram matrix and kernel derivatives

For each particle index pair $(n, m) \in \{1, \dots, N\} \times \{1, \dots, N\}$:

1. Evaluate the kernel $k(\boldsymbol{\theta}_n, \boldsymbol{\theta}_m)$
2. Evaluate the kernel gradient $\boldsymbol{\nabla}_{\boldsymbol{\theta}_n} k(\boldsymbol{\theta}_n, \boldsymbol{\theta}_m)$

Step 3: Approximate ensemble-based Jacobian

For particle n **from** 1 **to** N:

1. Calculate $\boldsymbol{\nabla}_{\boldsymbol{\theta}_n} \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)$ (see Section 3.2)

Step 4: Determine gradient descent direction

For particle n **from** 1 **to** N:

1. Calculate direction $\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)$ (see Equation 13)
2. Normalize it to obtain $\overline{\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)}$

Step 5: Identify gradient similarity

If $i > 1$:

For particle n **from** 1 **to** N:

1. Calculate acceleration proposal $a_{i,n}$ (see Equation 24)

Step 6: Adjust gradient descent step size

1. Average acceleration proposals to get $\overline{a_i}$
2. Adjust step size $\varepsilon_i = \varepsilon_{i-1} \overline{a_i}$ (see Equation 25)

Step 7: Apply gradient descent

For particle n **from** 1 to N:

1. Check for limits of $\boldsymbol{\theta}_{i,n} + \varepsilon_i \boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)$, adjust $\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)$ if required
2. Update particles $\boldsymbol{\theta}_{i+1,n} = \boldsymbol{\theta}_{i,n} + \varepsilon_i \boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)$ (see Equation 1)

Step 8: Check for convergence

If convergence criterium fulfilled:

Set $iterate = False$

Figure 3. Pseudo-code of the SVGD algorithm used in this study. Step 3 can be replaced if other methods for obtaining the Jacobian are available, Steps 4.2 to Step 6 may be replaced if a different Gradient Descent method is used.

4 Synthetic test case

4.1 Setup

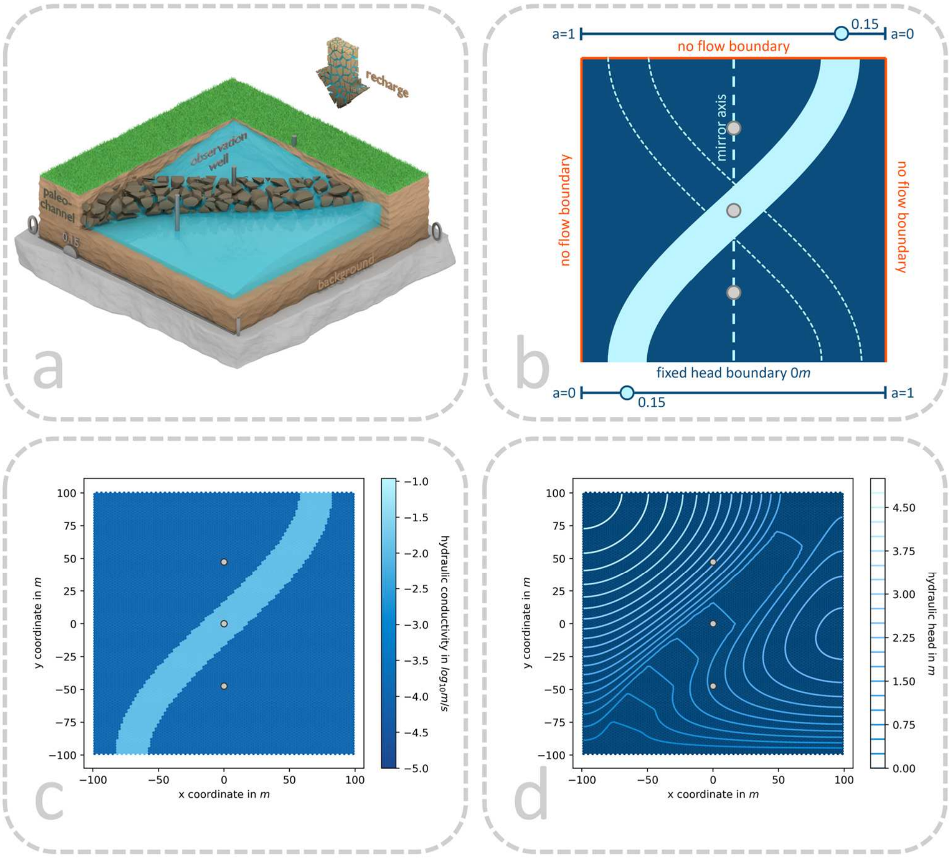


Figure 4. Conceptual render (a), conceptual sketch (b), true hydraulic conductivity field (c), and resulting true hydraulic head field (d) of the simple synthetic test case.

To illustrate the practical capabilities of SVGD, we first consider a simple synthetic test case. Towards this end, we construct a numerical hydrogeological model with a single parameter informing the uncertain path of a high-conductive paleo-channel in a two-dimensional, unconfined setting. This setup is illustrated in Figure 4. The system is defined as steady-state. Flow is driven by uniform recharge of 10^{-7} m/s over the model domain and drains to the southern fixed-head border. All other borders are assumed no-flow. Hydraulic conductivities of the background and paleo-channel are defined as 10^{-5} m/s and 10^{-3} m/s, respectively. Specific yield was set to $S_y = 0.15$, and the top and bottom elevation of the aquifer were set to 10 m and -10 m. The model parameter $0 < a < 1$ defines the start- and endpoint of a spline tracing the paleo-channel. The true solution is assumed to be $a = 0.15$, and the prior is defined as a beta distribution with parameters $\alpha, \beta = 2$. Observations are collected in three wells along the central north-south axis with an observation standard deviation of $\sigma = 0.025$ m. The model is implemented in MODFLOW 6 (Langevin et al., 2017) using the Python interface FloPy (Bakker et al., 2016).

We would like to draw attention to the fact that the setup of this scenario is symmetric with respect to the central north-south axis. As such, we would expect that there are two functionally indistinguishable solutions to the inference problem: $a = 0.15$ and $a = 0.85$. We test the algorithm with an ensemble of $N = 100$ particles, 100 iterations, an initial step-size of $\varepsilon_{i,0} = 10^{-4}$, an acceleration rate of $\alpha = 1.5$, and a similarity cutoff of $\beta = 0.75$. The kernel bandwidth was set to the mean distance to the $k = 25^{\text{th}}$ nearest neighbor during each iteration.

4.2 Results

Results of the inference process are illustrated in Figure 5. The posterior parameter field (Figure 5a, b) reveals that the expect bimodal uncertainty structure was successfully recovered by the algorithm: roughly half the ensemble places the channel at $a = 0.15$, the other half at $a = 0.85$. If this were a real scenario, this ambiguity could be resolved with additional geological information, or a new observation well located to the left or right of the mirror axis.

To test if the algorithm truly converged against the posterior, we compare the posterior ensemble against results obtained from a emcee (Foreman-Mackey et al., 2013) chain (Figure 5g, background). The emcee chain was obtained with 100 walkers and 445 jumps each, after removing the burn-in. Figure 5f verifies that SVGD seems to not only identified the correct posterior location, but also its spread.

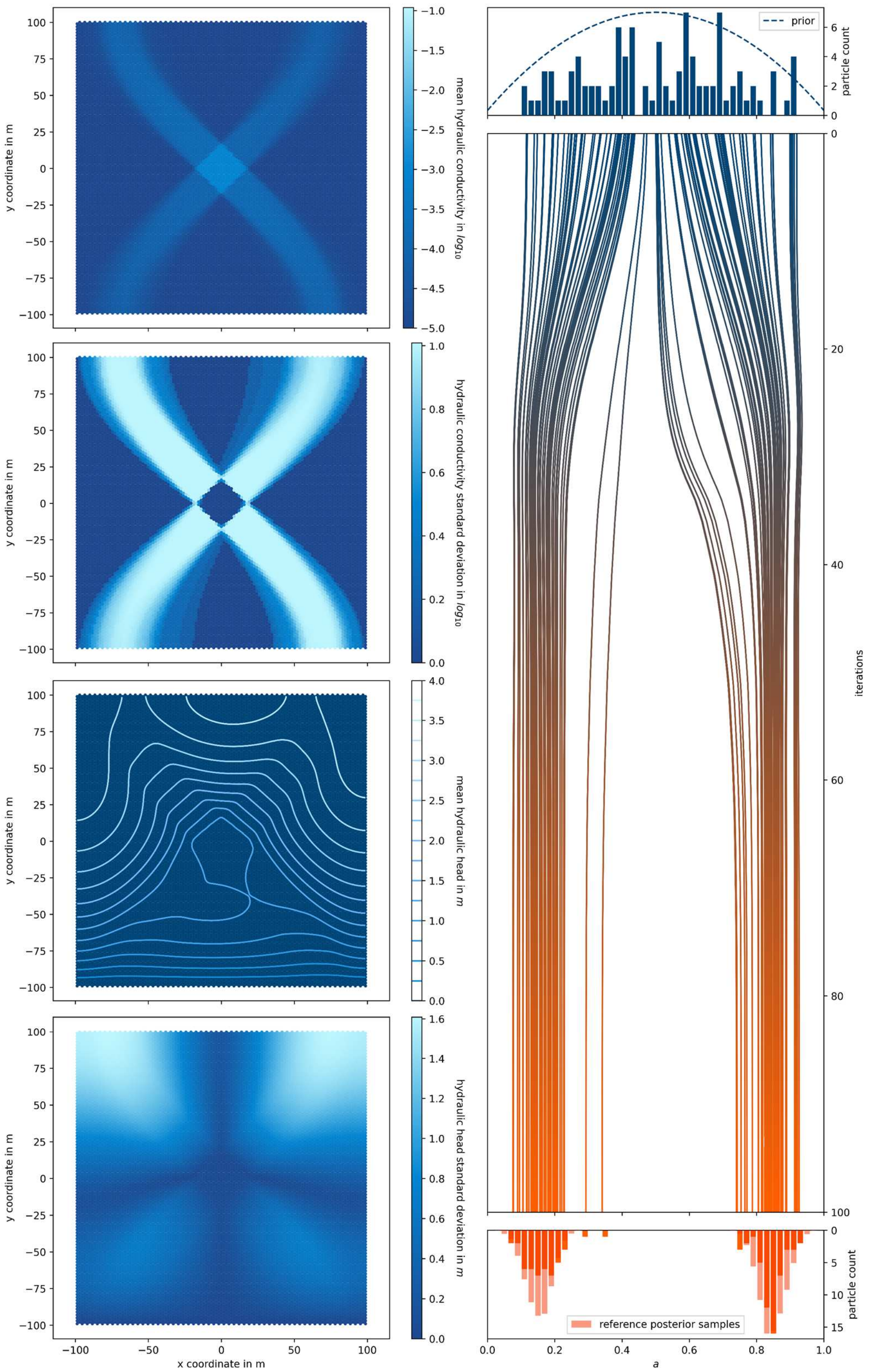


Figure 5. Results for the SVGD algorithm applied to the simple model: the left column shows the mean and standard deviation of hydraulic conductivity (a, b) and simulated head (c, d) at the end of the inference process. The right column illustrates the prior ensemble (e), the particle trajectories through the iterative process (f), and the resulting posterior ensemble (g).

5 Case study

5.1 Site description

For the real test case, we focus on the Kempt valley in Switzerland, a small pre-alpine catchment located about 10 km east of the city of Zurich. Within the valley lies the city of Fehraltorf, surrounded by pastures. The valley is characterized as follows:

- **Geology**: The aquifer layout is highly heterogeneous, shaped by alpine geology and postglacial sedimentology. Multiple electric resistivity tomography campaigns failed to delineate the aquifer bottom, and the prevailing gravelly sediments preclude direct push coring past a depth of approximately 7 m. Geological maps and indirect information suggest north-eastern and south-western plateaus or banks or impermeable material (Figure 6a).
- **Hydro(geo)logy**: The groundwater table is generally shallow, sometimes ponding during spring or after large precipitation events. Consequently, large swathes of the valley are artificially dewatered with tile drainages. The central Kempt stream is only perennial past the city of Fehraltorf, where it is sustained by a local wastewater treatment plant (WWTP), drainage channels, and multiple culverted creeks (Vögeli, 2018). Upstream of Fehraltorf, the creek is called Luppmen and controlled almost exclusively by groundwater. The groundwater table in the catchment is highly seasonally variable, particularly during the simulated drought year of 2018.
- **Infrastructure**: Due to the shallow groundwater table, the urban drainage network beneath Fehraltorf (Figure 6c) is partially submerged and substantial groundwater infiltration is known to occur. We further know the extraction rates for two municipal pumping stations near the southern edge of the city (Figure 6e). The agricultural estates in the catchment and an industrial greenhouse vegetable farm have concessions for ground- and river water extraction, but unfortunately no quantitative rates were available for either. Consequently, we neglected these potential sinks in the model..

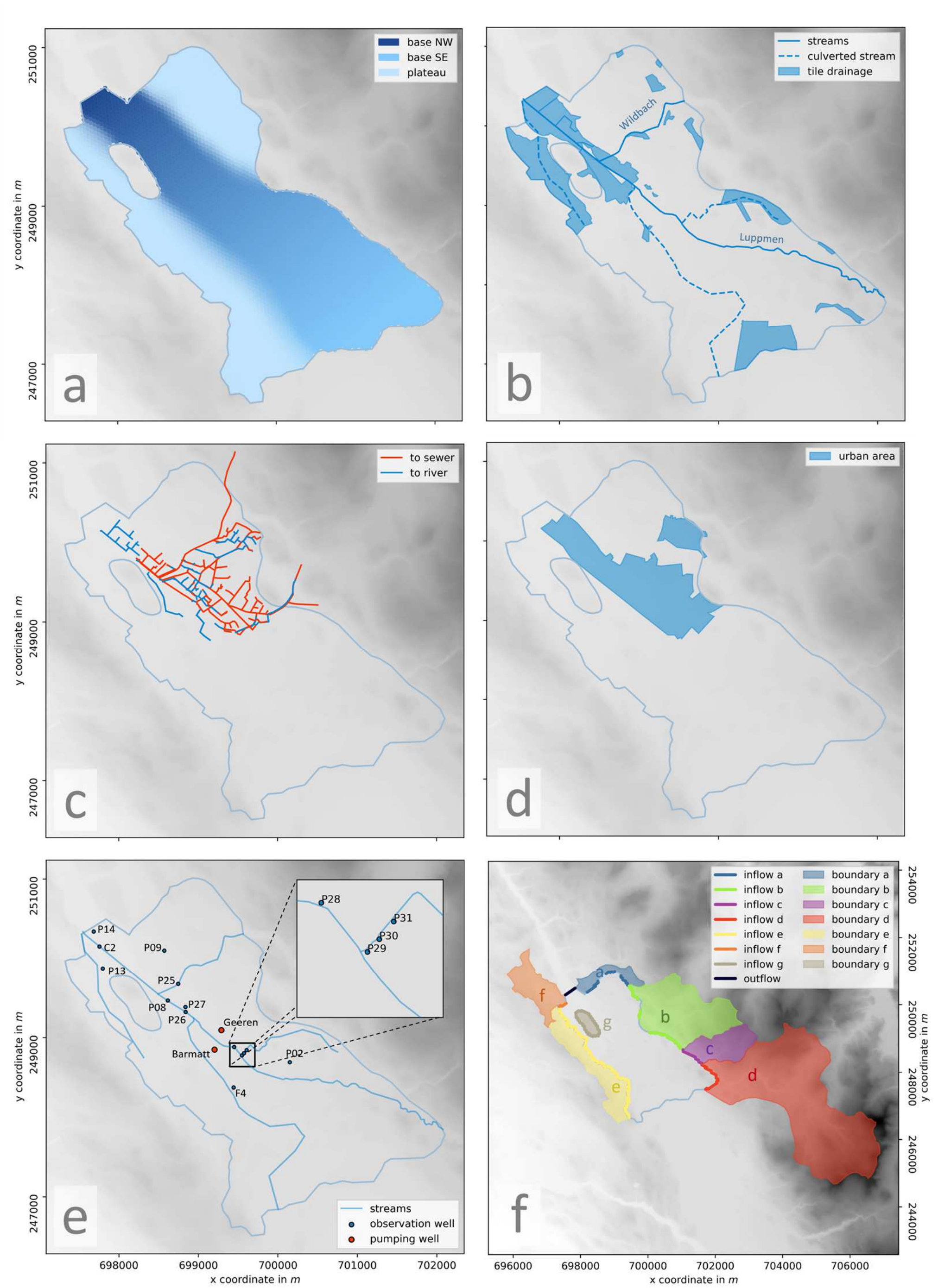


Figure 6. Approximate aquifer topology (a), tile drainage, open and culverted streams (b), extent of urban drainage network (c) and urban area (d), location of pumping and observation wells (e), and upslope contributing areas (f).

- **Boundary conditions**: Located in a headwater catchment, we expect that the valley receives significant inflow from the surrounding hillslopes (Figure 6f). We did not explicitly simulate these hillslopes, instead delineating six upslope catchments based on topographic information. These upslope catchments form the basis of conceptual models with uncertain extent and temporal dynamics which define the time-variable inflow into the central model. Vertical recharge is applied without delay and estimated from the difference between precipitation measurements within Fehraltorf and spatially averaged measurements of actual evapotranspiration in surrounding stations.

We simulate the drought year of 2018 (Bader et al., 2018), during which groundwater extraction and use had to be restricted due to an exceedingly low water table. We initialize the model with a seven-month spin-up period starting June 1st 2017, following a steady-state simulation with average meteorological conditions. We assimilate hydraulic head data from a number of observation wells (Figure 6e) as well as estimates of sewer infiltration rates, i.e. groundwater entering the sewers through deteriorated pipes and joints, obtained by a flow component separation (Becker et al., 2012) of distributed in-sewer flow rate measurements (Blumensaat et al., 2020a).

5.2 Model setup

We implement the numerical model in MODFLOW 6 (MF6: Langevin et al., 2017) using FloPy (Bakker et al., 2016). This framework permits a Newton-Raphson formulation for unstructured grids, which is more resilient to the drying of model cells. Furthermore, its modular structure and mover package permit the representation of the complex interactions of the stream, canalization, drainage system, and groundwater. Capitalized three-letter acronyms in the following paragraph refer to the respective MF6 packages.

We tessellated the model domain with a single layer of 4079 hexagonal prisms. The depth of the aquifer bottom is defined by four parameters which specify the elevation of four masks: the northwest-to-southeast gradient, the north-eastern plateau, and the south-western plateau (Figure 6a). Hydraulic conductivity is extrapolated through inverse distance weighting (Shepard, 1968) from 30 nodes. Tile drainages, the culverted creeks, and the urban drainage network are implemented as drainage elements (DRN), whose flows are diverted to their respective outflow points in the Luppmen through the mover (MVR) package. The conductance of the sewer pipes (hydraulic conductivity × cross-sectional area ÷ thickness) is extrapolated from ten nodes, and implemented as a 'pre-conductance' (hydraulic conductivity ÷ thickness), to be multiplied by the sewer pipes' surface area in order to yield the element's conductance. Where streams (Figure 6b) are open, their bed elevation has been measured, where they are culverted, their elevation has been extrapolated. The tile drainages were assumed to be located 0.75 m below the surface. The two non-culverted streams, Luppmen (main stream from SE to NW, Figure 6b) and Wildbach (northernmost stream, from NE into Luppmen, Figure 6b) are represented with the surface flow routing (SFR) module, which permits exchange with groundwater in both directions. The riverbeds' hydraulic conductivity was set to 10^{-5} m/s, the riverbed thickness to 30 cm, and its width to 3 m (Luppmen) or 1.5 m (Wildbach). Their Manning's coefficients are adaptable parameters. Direct runoff due to surface sealing in the urban areas is represented through a 35% flat recharge reduction. Infiltration into the sewer pipe network is consider in two ways: infiltration into storm sewers, and infiltration into the combined sewer system. The former is routed directly into adjacent surface waters (small creeks and the river Luppmen). The latter is used for inference against an estimated fraction of the total wastewater treatment plant (WWTP) inflow. The total WWTP outflow (in terms of volume balancing essentially the same as the WWTP inflow) – simulated groundwater infiltration plus domestic wastewater component – is routed into the Luppmen.

Recharge is estimated from the difference of average precipitation measurements around Fehraltorf (Blumensaat et al., 2020b) and a spatially averaged evaporation estimate from Meteoswiss (2020). Since the groundwater table is shallow and the time steps are coarse – set to three hours each –, we assumed instantaneous recharge within the valley. Recharge on the hillslopes is routed into the valley through time-variable inflow boundaries (Figure 6f) according to a simple, conceptual forcing model (Figure 7). This forcing model multiplies each timestep's raw recharge estimate with each boundary's upslope area (delineated based on topography) and a recharge multiplier. The latter is intended to compensate for potential deviations of the unknown groundwater catchment from the topographic catchment, bias in the recharge estimate, as well as unknown sinks or sources along the hillslopes. The resulting volumetric flux is then distributed among the subsequent timesteps according to an exponential distribution, whose extent is defined by a second parameter, the recharge delay. This parameter is intended to represent unresolved surface- and groundwater flow processes along the hillslope and controls the flashiness of the inflow. The temporally distributed volumetric flux components are then added to a new volumetric flux time series, and the process is repeated for the next time step. Once the new time series is assembled, the volumetric fluxes are distributed spatially across the respective boundary's inflow cells (Figure 6f).

In total, the numerical model features $D = 61$ parameters, some of which (hydraulic conductivity nodes, aquifer bottom elevation nodes, and forcing model parameters) are first converted into grid parameters using deterministic pre-processors. The priors of the parameters are illustrated in Table 1.

5.3 Algorithmic setup

We test the SVGD algorithm with two different ensemble sizes: an ensemble size of $N = 30$ and an ensemble size of $N = 100$. Considering the parameter space dimensionality of $D = 61$, the former scenario is restricted to exploring a subspace, while the latter scenario should have access to full parameter space. Consequently, we will focus on the $N = 100$ in the discussion of the results, as this scenario avoids the risk of misinterpreting optimization results. In both scenarios, we iterated 100 times. The required simulation time was about 30 hours for the $N = 30$ scenario, and about 102 hours for the $N = 100$ scenario.

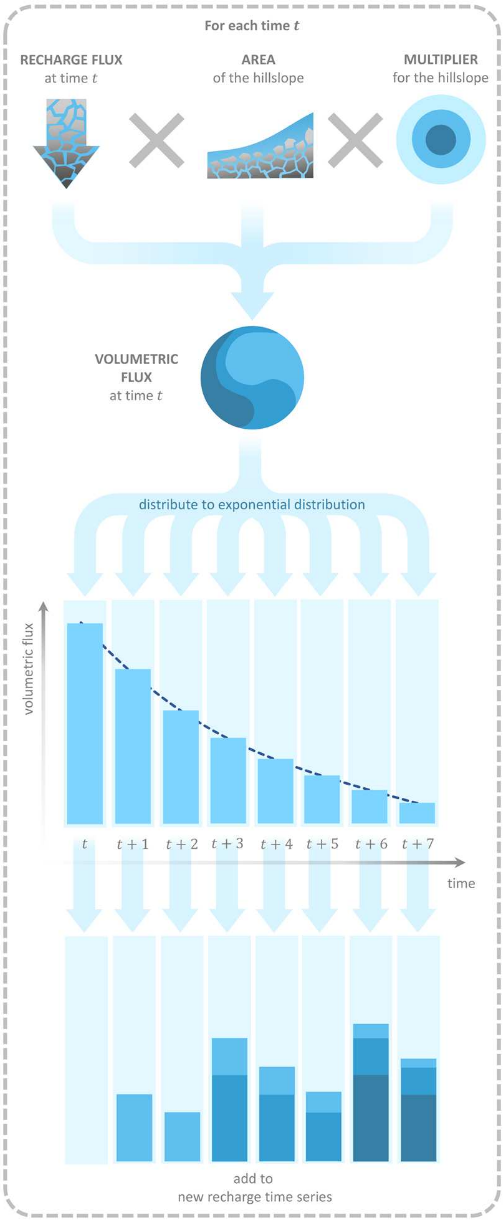
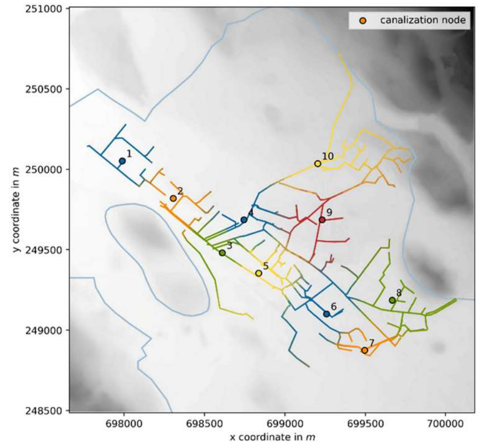


Figure 7. Illustration of the forcing model. For each time step and boundary, recharge estimates are multiplied by its area and a multiplier. The resulting volumetric flux is then distributed to subsequent timesteps according to an exponential distribution scaled by a recharge delay parameter. Finally, the distributed fluxes of each time are added up to yield the volumetric boundary inflow, distributed across its inflow model cells.

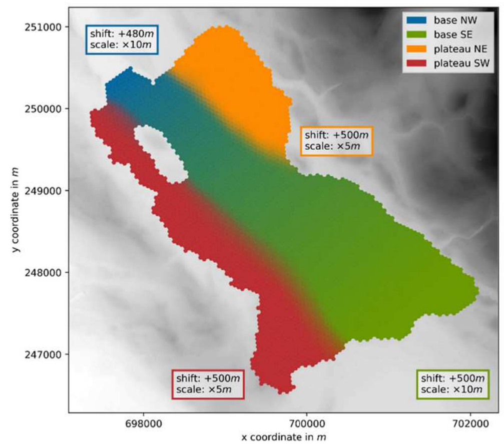
Table 1. Model parameters, priors and limits. Capitalized letters in the note column correspond to boundaries, LP refers to Luppmen, WB refers to Wildbach. Colored regions in map 1 and map 3 illustrate influence areas of different nodes. The ring above the north-eastern plateau in map 2 marks the mean of its slope orientation.

parameter		prior			limits		transformation	
name	note	pdf type	μ or α	σ or β	min	max	shift	scale
recharge delay	A-F	beta	5	7	+0.01	+0.99	+0	× 3
recharge multiplier	A-F	normal	0	1	−5	+5	+0	× 0.05
river flow fraction	B, D	beta	3	3	+0.01	+0.99	+0	× 1
Manning's coefficient	LP/WB	beta	3	5	+0.01	+0.99	+0.01	× 0.08
canalization pre-conductance	map 1	beta	2	5	+0.01	+0.99	−7	× 7
aquifer bottom elevation	map 2	normal	0	1	None	None	see map 2	see map 2
hydraulic conductivity	map 3	normal	0	1	−3	+3	−4	× 0.5
specific yield	None	beta	5	15	+0	+1	+0	× 0.5

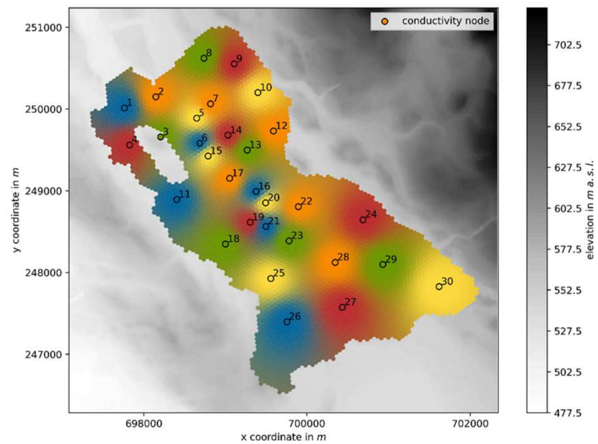
map 1: *interpolation of canalization pre-conductance*



map 2: *interpolation and shift/scale of aquifer topology*



map 3: *interpolation of hydraulic conductivity*



5.4 Results

The simulated states at the observation wells and the urban drainage network for the posterior ensemble of the $N = 100$ scenario are illustrated in Figure 8, for the prior ensemble and the scenario $N = 30$ in Figure S1 and Figure S2 (supporting information). Improvements to the simulated hydraulic heads are significant, reducing the root mean square error (RMSE) from a prior average of 312 cm down to a posterior average of 30 cm (Figure 9) for the scenario $N = 100$, and from 322 cm down to 39 cm for the scenario $N = 30$ (Figure S2). Proportionally, bias is reduced even further, from a prior mean of 207 cm down to a posterior mean of only 4 cm in the case of $N = 100$, and from 201 cm to 2 cm for $N = 30$. The slightly elevated RMSE contrasted by very low bias suggests that the residual error is rooted in model structural deficiencies.

We expect a significant impact from such model deficiencies since we only employed a single prescribed head boundary at the outflow. Consequently, all hydraulic head fluctuations within the model domain must be created by the model itself, instead of being partially inherited from the dynamics of a hypothetical upslope prescribed head boundary. The simulated and observed hydraulic heads seem to support this interpretation (Figure 8). The model successfully recreated the yearly dynamics in most wells, but we can observe varying patterns between them, often with errors which may have a plausible model-structural explanation:

Observations at wells C2 and P08 (Figure 8a and d), for example, barely fluctuate over the year and retain a relatively steady water level. This suggests that both wells are subjected to some form of stabilizing influence, likely a perennial drainage effect. Both wells are located adjacent to the river Luppmen and the urban drainage network, and their similar elevation – matching the observed water tables – makes either feature a plausible stabilizing influence.

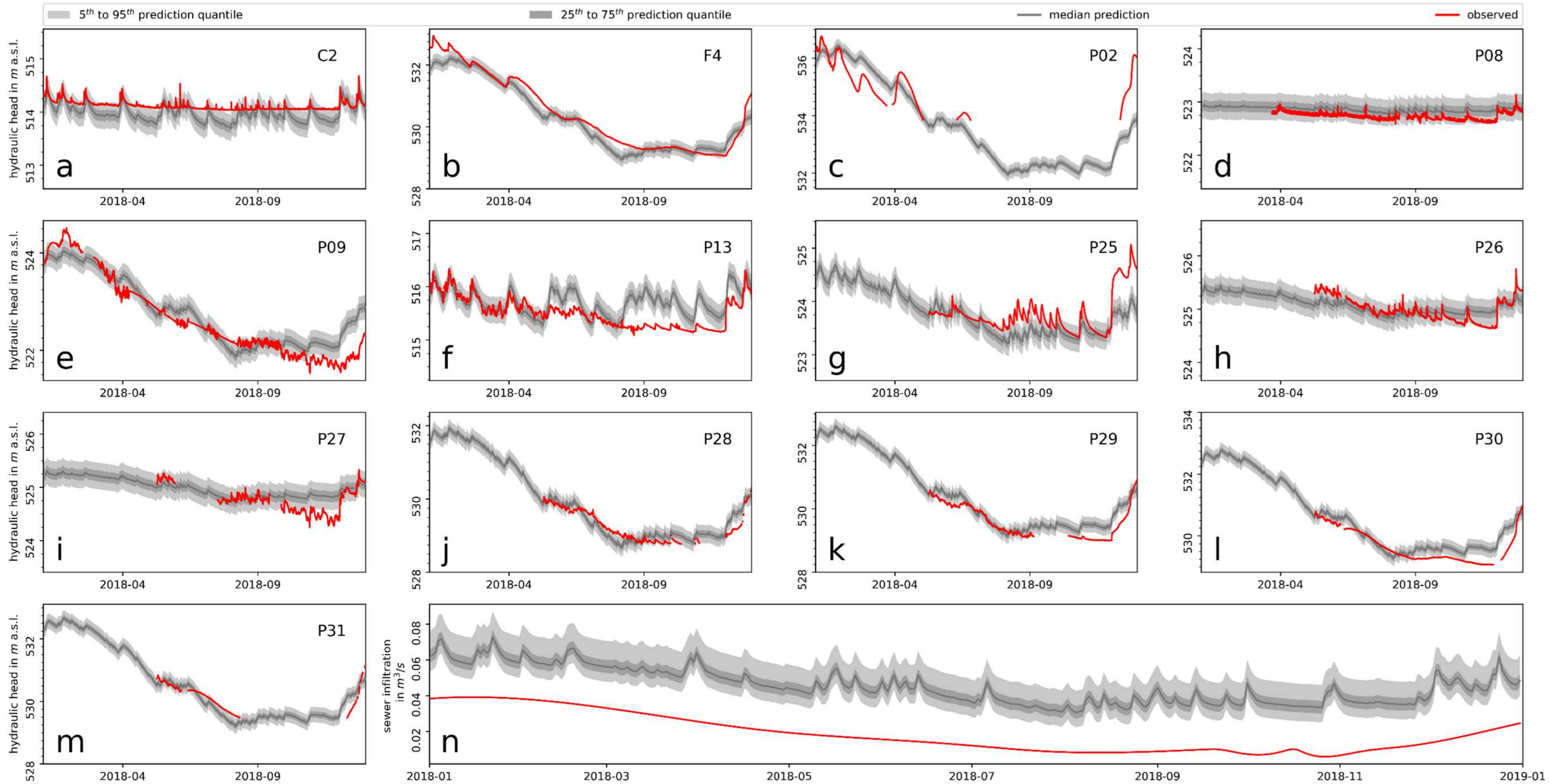


Figure 8. Posterior simulated (greyscale) and observed (red) hydraulic heads (a-m) and canalization groundwater infiltration (n) with model error at the end of simulation period for $N = 100$. Prior results are illustrated in Figure S3.

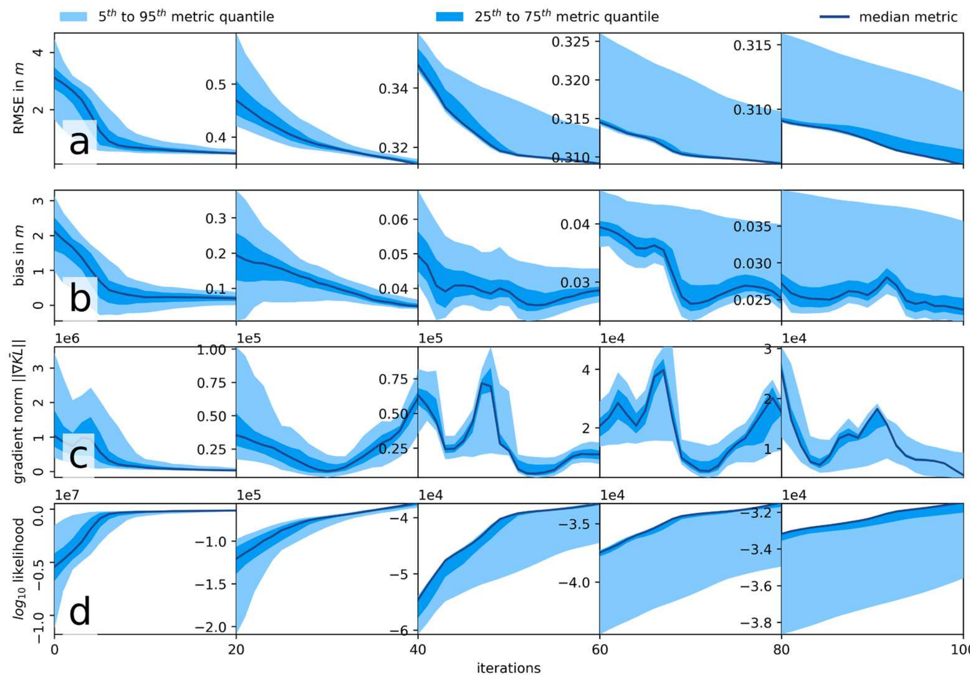


Figure 9. Posterior overall root-mean square error (o) and bias (p) for the hydraulic heads, the mean norm of the Kullback-Leibler divergence gradient (q) and the log-likelihood (r) across the algorithm's iterations. For better visualization, the y-axis scale is reset every 20 iterations for the scenario $N = 100$.

The inference favored increasing the canalization's leakiness near both wells (Figure 10g), possibly because the Luppmen's riverbed conductance was assumed spatially uniform and hence did not allow for local adjustments. However, groundwater infiltration into the sewer network (Figure 8n) is consistently overestimated – compared to estimations based on in-sewer flow observations –, which could suggest that the river has a larger role to plays in the stabilization of P08 and C2.

Wells F4, P02, P09, and P28 to P31 (Figure 8b, c, e, j-m) feature similar yearly trends, recovered to varying degrees of fidelity: A steady water table decrease by up to 3 m from January to September, followed by a steep rebound in late autumn. While the water table drop is reproduced faithfully, its rebound is underestimated in all wells. A possible explanation is the omission of agricultural water extraction. The rebound in autumn is likely a composite effect of direct recharge and the deactivation of irrigation systems, the latter of which is unrepresented in the model. This suggests the model compensated for the omission of agricultural extraction during the main vegetation period through other means.

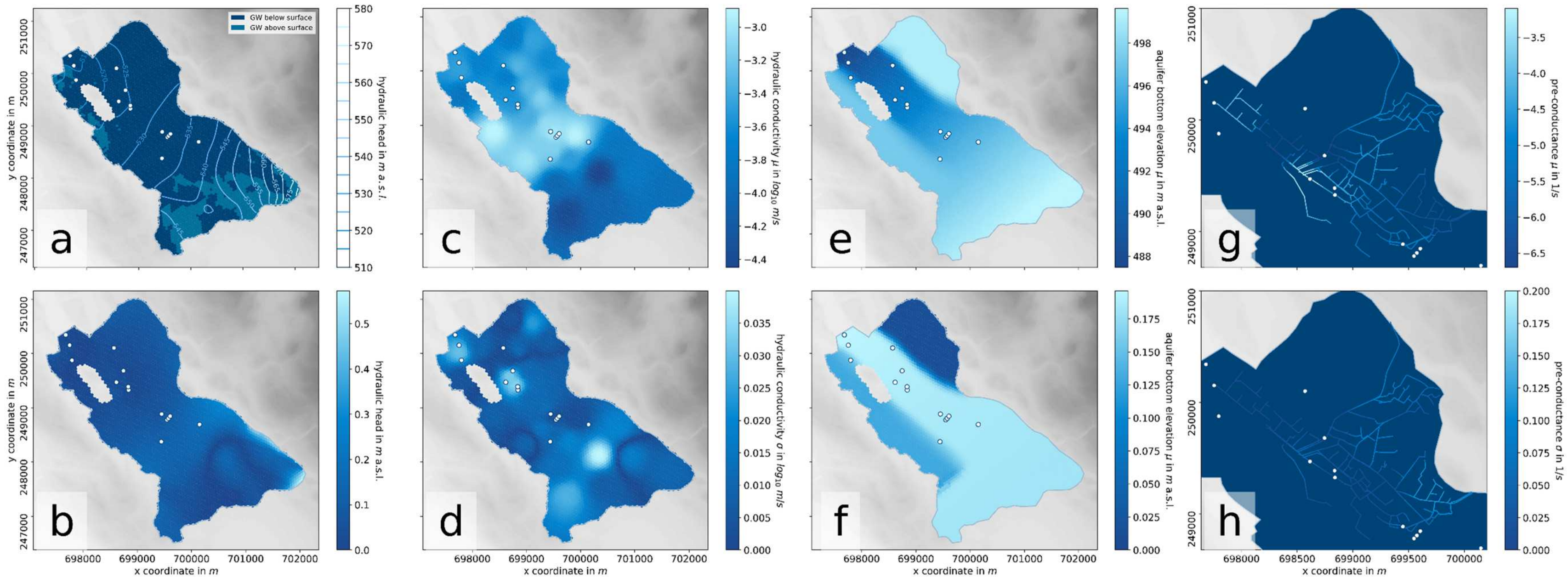


Figure 10. Posterior parameters and hydraulic head at the final iteration for $N = 100$. The two rows illustrate mean (a, c, e, g) and standard deviations (b, d, f, h) of hydraulic head in the initial steady-state simulation period (c, d), hydraulic conductivity (e, f), aquifer bottom elevation (g, h), and canalization conductance (i, j). Recharge parameters are illustrated in Figure S5, and the corresponding prior fields are illustrated in Figure S6 and Figure S7. Results for the scenario $N = 30$ are shown in Figure S8 to S11.

The patterns in the remaining wells are somewhere between the two sets discussed above. P13 (Figure 8f) is located in an agricultural area with tile drainages and diverges from the observed water tables only from May onwards. The remaining wells (P25-P27, Figure 8g-i) are located in the urban area of Fehraltorf and feature fluctuations which the model cannot seem to fully recreate.

Overall, it seems our prior parameter assumptions resulted in an initial overestimation of water tables, which SVGD corrected by reducing hydraulic conductivities (Figure 10c) relative to the prior, particularly near the centre of the catchment. Individual changes to parameter uncertainty for the scenarios $N = 100$ and $N = 30$ are illustrated in Figure S12 and Figure S13 (supporting information). The model simulates groundwater ponding in the initial steady-state spin-up period near the southern and western edges of the valley (Figure 10a). This may not be unrealistic, as both areas feature tile drainages, which indicate historical issues with ponding groundwater. Particularly in the southern region we have some evidence for ponding: a naturally marshy, extensively drained forest, and a small airfield whose runway is often closed during spring due to swampy grassland conditions.

6 Discussion

In summary, the inference results of SVGD were promising, returning the true posterior in the synthetic test case, and yielding substantial improvements in terms of predictive error for the application to a truly complex case study. In the latter scenario, the observed states did not always remain within the error bounds, which suggests both structural model inadequacy and an underestimation of the model error. We identified some potential sources of this error – the omission of agricultural irrigation, and imperfect representation of canalization and riverbed drainage –, which could be revised in a future iteration of the conceptual model. The standard deviation of the model error is a parameter which could also be inferred, although we note that this would complicate the derivative of the loglikelihood gradient (Equation 18). A further interesting addition would be the consideration of temporal correlation in the model error covariance matrix, which may prevent the strong tapering of the posterior in the real test case.

As far as the inference itself is concerned, SVGD successfully recovered the synthetic bimodal posterior – a nigh-impossible task for non-localized methods based on the assumption of Gaussianity. In the real test case, no exhaustive reference solution was available. Results for the subspace-limited $N = 30$ scenario were promising. While we acknowledge that it is undesirable and potentially dangerous to restrict parameter inference to a subspace, computational limitations often demand working within such restrictions. This ability to recover at least simplified uncertainty estimates in settings with inevitably insufficient computational resources constitutes, in our opinion, one of the main advantages of the EnKF and is shared by our implementation of SVGD.

Despite the promising optimization performance, this algorithm comes at a computational price: the necessity to iterate requires re-simulating the full observation history for each particle during every iteration, whereas filter methods like the EnKF must only simulate the model history once for each particle (albeit separated into distinct assimilation time steps). However, it may not be necessary to iterate for as long as we did in our test cases – towards the end of the iteration period, improvements were only minor. We remain confident that performance can be improved further with adjustments to the gradient descent algorithm.

7 Conclusions

In this study, we employed the Stein Variational Gradient Descent algorithm of Liu & Wang (2016) and proposed adaptations for its practical application to non-Gaussian parameter inference in hydrogeological models. Towards this end, we proposed a computationally inexpensive, localized, ensemble-based approximation of the Jacobian. This matrix is used for the calculation of the logposterior gradient, and possibly the greatest computational obstacle to the implementation of SVGD. We also proposed a simple gradient descent algorithm which optimizes the algorithm's computational efficiency by adapting the descent step size dynamically.

We then proceeded to illustrate the performance of the algorithm in two test cases: a simple synthetic model with an intuitive solution, and a complex model based on a real field site with non-trivial, nonlinear parameter interactions. Results in both cases were promising. Our application in the synthetic test case successfully converged against the bimodal reference solution obtained by MCMC, iteratively evolving a unimodal prior into a bimodal posterior. While no reference solution was available for the real test case, inference results seem promising as well. Despite the model's complexity, the inference significantly reduced simulation error and bias, with the residual error likely being based on model structural inadequacy. Throughout, the algorithm retained uncertainty without the need for artificial variance inflation, a challenge for particle filters (e.g., Ramgraber et al., 2019, 2020) or the EnKF (Anderson, 2007). We tested the algorithm's performance (and the fidelity of our approximations) for two ensemble sizes: $N = 30$, restricted to an at most 29-dimensional parameter-subspace, and $N = 100$, with theoretical access to all parameter space dimensions.

Substantial improvements were obtained in both scenarios, although the larger ensemble size yielded slightly better optimization results.

A limitation of this algorithm is its restriction to smooth probability distributions with at least convex support, a weakness shared with other gradient descent algorithms and the EnKF. For the inference of structural uncertainty of geological facies, it may be necessary to employ an auxiliary parameterization which permits a smooth or convex supported pdf first (e.g., Hu et al., 2013; Ramgraber et al., 2019). A further possible source of error may be found in our ensemble-based Jacobian approximation. While our synthetic example converged successfully and optimization results were promising in both test cases, we cannot guarantee that this approximation proves adequate in all cases.

For future research, we are optimistic that the experimentation with other gradient descent algorithms could improve the efficiency of the SVGD algorithm even further. Alternative Jacobian approximations, particularly those obtained with automatic differentiation, seem a promising way to improve the fidelity of practical applications of SVGD and constitute an important avenue for future research. Alternatively, using the unnormalized logposterior estimates at the particles to approximate the logposterior gradient directly could also be an interesting research direction. Other fascinating research directions could be found in the related field of transport maps (e.g., Marzouk et al., 2017; El Moselhy & Marzouk, 2012; Spantini et al., 2018) which construct the transformation functions explicitly. In conclusion, we believe that SVGD is a highly promising and relatively easy-to-use (although not necessarily easy-to-derive) tool for non-Gaussian parameter inference in hydrogeological systems, and that a strong case could be made for its use in complex models with weak claim to Gaussianity.

8 Acknowledgements

We express our sincere gratitude to Dr. Carlo Albert, Swiss Federal Institute of Aquatic Science and Technology (Eawag), for many discussions and guidance during the interpretation and re-derivation of the SVGD algorithm. We furthermore want to thank Prof. Manuel Pulido, University of Reading, for providing the source code of his publication's examples, which aided the creation of our algorithm. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675120. The data and codes accompanying this manuscript are available under: non-existent dummy DOI, but temporarily at https://drive.switch.ch/index.php/s/r30p3eL0jCApFTf

9 References

Anderson, J. L. (2007). An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus, Series A: Dynamic Meteorology and Oceanography*. https://doi.org/10.1111/j.1600-0870.2006.00216.x

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian\nBayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188. https://doi.org/10.1109/78.978374

Bader, S., Burgstall, A., Casanueva, A., Duguay-Tetzlaff, A., Gehrig, R., Gubler, S., et al. (2018). *Hitze und Trockenheit im Sommerhalbjahr 2018*. Zürich. Retrieved from https://www.meteoschweiz.admin.ch/content/dam/meteoswiss/de/service-und-publikationen/Publikationen/doc/Fachbericht_TrockenheitHitze_2018_final_d.pdf

Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J., & Fienen, M. N. (2016). Scripting MODFLOW Model Development Using Python and FloPy. *Groundwater*, *54*(5), 733–739. https://doi.org/10.1111/gwat.12413

Becker, M., Brombach, H., Hennerkes, J., Holte, A., Jütting, F., Nebauer, M., et al. (2012). *Fremdwasser in Entwässerungssystemen außerhalb von Gebäuden*. 53773 Hennef, Germany.

Bengtsson, T., Bickel, P., & Li, B. (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and Statistics: Essays in Honor of David A. Freedman*. https://doi.org/10.1214/193940307000000518

Betancourt, M. (2018). A Conceptual Introduction to Hamiltonian Monte Carlo. arXiv. Retrieved from https://arxiv.org/abs/1701.02434

Blumensaat, F., Dicht, S., Disch, A., & Maurer, M. (2020a). The Urban Water Observatory - Long-term monitoring of urban water resources dynamics in very high spatiotemporal resolution using low-power sensor and data communication techniques. Urban Water Observatory. Retrieved from https://uwo-opendata.eawag.ch/

Blumensaat, F., Dicht, S., Disch, A., & Maurer, M. (2020b). The Urban Water Observatory - Niederschlagsintensität. Urban Water Observatory. Retrieved from https://uwo-opendata.eawag.ch/

Carrera, J., Alcolea, A., Medina, A., Hidalgo, J., & Slooten, L. J. (2005). Inverse problem in hydrogeology. *Hydrogeology Journal*. https://doi.org/10.1007/s10040-004-0404-7

Chen, Y., & Oliver, D. S. (2013). Levenberg-Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification. *Computational Geosciences*. https://doi.org/10.1007/s10596-013-9351-5

Chwialkowski, K., Strathmann, H., & Gretton, A. (2016). A kernel test of goodness of fit. In *33rd International Conference on Machine Learning, ICML 2016*.

Cirpka, O. A., & Valocchi, A. J. (2016). Debates—Stochastic subsurface hydrology from theory to practice: Does stochastic subsurface hydrology help solving practical problems of contaminant hydrogeology? *Water Resources Research*. https://doi.org/10.1002/2016WR019087

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, *99*(C5), 10,110-143,162. https://doi.org/10.1029/94JC00572

Evensen, G. (2003). The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, *53*(4), 343–367. https://doi.org/10.1007/s10236-003-0036-9

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013). emcee : The MCMC Hammer . *Publications of the Astronomical Society of the Pacific*. https://doi.org/10.1086/670067

Gu, Y., & Oliver, D. S. (2007). An Iterative Ensemble Kalman Filter for Multiphase Fluid Flow Data Assimilation. *SPE Journal*, *12*(4), 438–446. https://doi.org/10.2118/108438-pa

Hendricks Franssen, H. J., Kaiser, H. P., Kuhlmann, U., Bauser, G., Stauffer, F., Mller, R., & Kinzelbach, W. (2011). Operational real-time modeling with ensemble Kalman filter of variably saturated subsurface flow including stream-aquifer interaction and parameter updating. *Water Resources Research*. https://doi.org/10.1029/2010WR009480

Hu, L. Y., Zhao, Y., Liu, Y., Scheepens, C., & Bouchard, A. (2013). Updating multipoint simulations using the ensemble Kalman filter. *Computers & Geosciences*, *51*, 7–15. https://doi.org/10.1016/j.cageo.2012.08.020

Keller, J., Hendricks Franssen, H. J., & Marquart, G. (2018). Comparing Seven Variants of the Ensemble Kalman Filter: How Many Synthetic Experiments Are Needed? *Water Resources Research*. https://doi.org/10.1029/2018WR023374

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Langevin, C. D., Hughes, J. D., Banta, E. R., Niswonger, R. G., Panday, S., & Provost, A. M. (2017). Documentation for the MODFLOW 6 Groundwater Flow Model. *U.S. Geological Survey*. https://doi.org/10.3133/tm6A55

van Leeuwen, P. J. (2009). Particle Filtering in Geophysical Systems. *Monthly Weather Review*. https://doi.org/10.1175/2009MWR2835.1

van Leeuwen, P. J., Künsch, H. R., Nerger, L., Potthast, R., & Reich, S. (2019). Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.3551

Linde, N., Renard, P., Mukerji, T., & Caers, J. (2015). Geological realism in hydrogeological and geophysical inverse modeling: A review. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2015.09.019

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

Liu, Q., & Wang, D. (2016). Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*.

Liu, Q., Lee, J. D., & Jordan, M. (2016). A kernelized Stein discrepancy for goodness-of-fit tests. In *33rd International Conference on Machine Learning, ICML 2016*.

Margossian, C. C. (2019). A review of automatic differentiation and its efficient implementation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. https://doi.org/10.1002/widm.1305

Marzouk, Y., Moselhy, T., Parno, M., & Spantini, A. (2017). Sampling via measure transport: An introduction. In *Handbook of Uncertainty Quantification*. https://doi.org/10.1007/978-3-319-12385-1_23

Meteoswiss. (2020). Meteoswiss. Retrieved from www.meteoswiss.admin.ch

Moeck, C., Molson, J., & Schirmer, M. (2020). Pathline Density Distributions in a Null-Space Monte Carlo Approach to Assess Groundwater Pathways. *Groundwater*. https://doi.org/10.1111/gwat.12900

Mongillo, M. (2011). Choosing Basis Functions and Shape Parameters for Radial Basis Function Methods. *SIAM Undergraduate Research Online*, *4*, 190–209. https://doi.org/10.1137/11s010840

Moradkhani, H., Hsu, K.-L., Gupta, H., & Sorooshian, S. (2005). Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resources Research*, *41*(5). https://doi.org/10.1029/2004WR003604

El Moselhy, T. A., & Marzouk, Y. M. (2012). Bayesian inference with optimal maps. *Journal of Computational Physics*. https://doi.org/10.1016/j.jcp.2012.07.022

Poeter, E., & Townsend, P. (1994). Assessment of Critical Flow Path for Improved Remediation Management. *Groundwater*. https://doi.org/10.1111/j.1745-6584.1994.tb00661.x

Pulido, M., van Leeuwen, P. J., & Posselt, D. J. (2019). *Kernel embedded nonlinear observational mappings in the variational mapping particle filter*. Retrieved from https://arxiv.org/abs/1901.10426

Ramgraber, M., Albert, C., & Schirmer, M. (2019). Data Assimilation and Online Parameter Optimization in Groundwater Modeling Using Nested Particle Filters. *Water Resources Research*. https://doi.org/10.1029/2018WR024408

Ramgraber, M., Camporese, M., Renard, P., Salandin, P., & Schirmer, M. (2020). Quasi-online groundwater model optimization under constraints of geological consistency based on iterative importance sampling. *Water Resources Research*. https://doi.org/10.1029/2019wr026777

Reichle, R. H., McLaughlin, D. B., & Entekhabi, D. (2002). Hydrologic Data Assimilation with the Ensemble Kalman Filter. *Monthly Weather Review*. https://doi.org/10.1175/1520-0493(2002)130<0103:hdawte>2.0.co;2

Renard, P. (2007). Stochastic hydrogeology: What professionals really need? *Ground Water*. https://doi.org/10.1111/j.1745-6584.2007.00340.x

Rubin, Y. (2003). *Applied stochastic hydrogeology*. Oxford University Press.

Sanchez-Vila, X., & Fernàndez-Garcia, D. (2016). Debates—Stochastic subsurface hydrology from theory to practice: Why stochastic modeling has not yet permeated into practitioners? *Water Resources Research*. https://doi.org/10.1002/2016WR019302

Schölkopf, B., & Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference on* -. https://doi.org/10.1145/800186.810616

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*. https://doi.org/10.1029/2007wr006705

Spantini, A., Bigona, D., & Marzouk, Y. (2018). Inference via low-dimensional couplings. *The Journal of Machine Learning Research*, *19*(1), 2639–2709.

Vögeli, E. (2018). *In Fehraltorf wird die Luppmen zur Kempt*. Fehraltorf. Retrieved from https://www.fehraltorf.ch/wAssets/docs/gemeinde/geschichte_chronik/verschiedenes_zur_dorfgeschichte/Luppmen-Kempt.pdf

Vrugt, J. A., ter Braak, C. J. F., Diks, C. G. H., & Schoups, G. (2013). Hydrologic data assimilation using particle Markov chain Monte Carlo simulation: Theory, concepts and applications. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2012.04.002

Wendt, J. F., Anderson, J. D., Degroote, J., Degrez, G., Dick, E., Grundmann, R., & Vierendeels, J. (2009). *Computational fluid dynamics: An introduction*. *Computational Fluid Dynamics*. https://doi.org/10.1007/978-3-540-85056-4

Werner, D. (2018). *Funktionalanalysis* (8th ed.). Berlin: Springer Spektrum. https://doi.org/https://doi.org/10.1007/978-3-662-55407-4

White, J. T. (2018). A model-independent iterative ensemble smoother for efficient history-matching and uncertainty quantification in very high dimensions. *Environmental Modelling and Software*. https://doi.org/10.1016/j.envsoft.2018.06.009