

Water Resources Research

RESEARCH ARTICLE

10.1029/2020WR029509

Key Points:

- In this study, we explore the use of the analytic element method (AEM) for hydrogeological uncertainty estimation using a Markov Chain Monte Carlo algorithm
- We include a flexible element based on conformal mapping for the influence of uncertain regional flow in a simple Python-based AEM toolbox
- We find that AEM can be a useful tool for direct steady-state uncertainty estimation or act as a support tool during model creation

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

M. Ramgraber,
max.ramgraber@eawag.ch

Citation:

Ramgraber, M., & Schirmer, M. (2021). Hydrogeological uncertainty estimation with the analytic element method. *Water Resources Research*, 57, e2020WR029509. <https://doi.org/10.1029/2020WR029509>

Received 22 DEC 2020
Accepted 19 MAY 2021

Hydrogeological Uncertainty Estimation With the Analytic Element Method

Maximilian Ramgraber^{1,2}  and Mario Schirmer^{1,2} 

¹Department of Water Resources and Drinking Water, Eawag—Swiss Federal Institute of Aquatic Science and Technology, Dübendorf, Switzerland, ²Centre for Hydrogeology and Geothermics (CHYN), University of Neuchâtel, Neuchâtel, Switzerland

Abstract Uncertainty estimation plays an important part in practical hydrogeology. With most of the subsurface unobservable, attempts at system characterization will invariably be incomplete. Uncertainty estimation, then, must quantify the influence of unknown parameters, forcings, and structural deficiencies. In this endeavor, numerical modeling frameworks can resolve a high degree of subsurface complexity and its associated uncertainty. Where boundary uncertainty is concerned, however, numerical frameworks can be restrictive. The interdependence of grid discretization and its enclosing boundaries render exploration of uncertainties in their extent or nature challenging. The analytic element method (AEM) may be an interesting complement, as it is computationally efficient, economic with its parameter count, and does not require enclosure through finite boundaries. These properties make AEM well suited for uncertainty estimation, particularly in data-scarce settings or exploratory studies. In this study, we explore the use of AEM for flow field uncertainty estimation, with a particular focus on boundary uncertainty. To induce diverse, uncertain regional flow more easily, we propose a new element based on a Möbius transformation. We include this element in a simple Python-based AEM toolbox and benchmark it against MODFLOW. Coupling AEM with a Markov Chain Monte Carlo routine using adaptive proposals, we explore its use in a synthetic case study. We find that AEM permits efficient uncertainty estimation for groundwater flow fields, which may form a basis for stochastic Lagrangian transport modeling or can support advanced model design by informing the placement of numerical model boundaries.

1. Introduction

Groundwater modeling plays an important role in practical hydrogeology. In a discipline in which neither the system nor its properties can be observed in its entirety, it is the task of models to establish spatial and temporal continuity between point-wise information. Where only few observations are available, uncertainty dominates the system characterization. These uncertainties must be quantified to endow any information derived from models with reliable confidence intervals.

Sources of model uncertainty are manifold and somewhat elusive, but arise from three main sources: unknown subsurface parameters (*parametric uncertainty*; e.g., Linde et al., 2017; Renard, 2007), unknown boundary conditions and forcings (*forcing uncertainty*; e.g., Guillaume et al., 2016; Vrugt et al., 2008), and model structural inadequacies (*conceptual uncertainty*; e.g., Höge et al., 2019). In practice, hydrogeologists tend to devote most resources to parametric uncertainty, focusing on the ambiguity in sediment parameters such as hydraulic conductivity. Where forcing uncertainty is considered, it is rarely explored beyond the addition of white noise to perturb model predictions. This omission can be problematic, since the influence of forcing uncertainty may be indistinguishable from the effects of parametric uncertainty (e.g., Erdal & Cirkpa, 2016) or even eclipse them (e.g., Peeters & Turnadge, 2019).

While all models of unisolated systems require boundary conditions in some form, the specific requirements depend on the chosen modeling framework. Numerical models place particularly stringent requirements, demanding a finite domain along which specified boundary conditions are enforced. Such boundaries are rarely well defined in reality. Common pragmatic choices are prescribed hydraulic head conditions inter- and extrapolated from marginal observation wells, or no-flow boundaries along anticipated streamlines. Since the nature of these boundaries intertwines them closely with the extent of the model grid, they do not lend themselves readily to the exploration of uncertainty in their extent or nature without adjusting the grid as well.

Prescribed in- or outflow boundary conditions would be a more versatile choice to represent the uncertain influence of regional flow in a finite domain, but are very difficult to obtain and consequently are rarely used. Recognizing this limitation, simulation frameworks such as MODFLOW 6 (Langevin et al., 2017) have since implemented multi-level setups which allow the use of simpler, large-scale models to define the flow boundaries of the main area of interest. This shifts the boundary problem up the hierarchy, but at additional computational expense.

In search of a remedy, an interesting alternative may be found in the analytic element method (AEM: Haitjema, 1995; Strack, 1989, 2003, 2017). Instead of discretizing the model domain into cells or finite elements, AEM constructs a complex-valued, scale-invariant analytic solution to the flow field for two-dimensional steady-state flow. This is achieved through superposition (addition) of simpler solutions, the eponymous analytic elements. While initially developed for two-dimensional, steady-state settings, AEM has since been extended to support three-dimensional model domains (Haitjema, 1985; Janković & Barnes, 1999), smooth inhomogeneities (Craig, 2009), and transient dynamics (Furman & Neuman, 2003). AEM code has been distributed in modeling frameworks such as TIMML (Bakker, 2006; Bakker & Strack, 2003), Visual AEM (Craig et al., 2009), TTim (Bakker, 2013), or AnAqSim (Fitts et al., 2015).

In the context of forcing uncertainty, AEM has the desirable property that it does not demand enclosure through finite, specified boundaries. Traditionally, regional flow is implemented as uniform and unidirectional flow, and subsequently deformed through the placement of elements outside the immediate domain of interest. In principle, this method can induce highly complex flow fields, but is not very well suited for practical uncertainty estimation due to its indirect nature. More substantial changes to regional flow would require changing the position and rotation of the farfield elements.

In this study, our objective is three-fold: we strive to (a) demonstrate the intrinsic suitability of AEM for Bayesian inference, capitalizing on its naturally low parameter count and computational efficiency. Its natural approach to complexity (*start simple, add more complexity as required*) can make it more suitable for exploratory analyses than numerical models, which are often explored the other way around (*start complex, then simplify by aggregating grid parameters*). Toward this end, we (b) propose a new element based on a Möbius transformation, which can directly induce curving, diverging, or converging regional base flow within a circular model domain of arbitrary size. This improves AEM's suitability for the exploratory analysis of boundary uncertainty. Finally, we (c) provide a simple modular Python toolbox to offer practitioners unfamiliar with AEM or Bayesian statistics a starting point for incorporating basic uncertainty estimation into their modeling workflow.

We subsequently demonstrate the use of AEM and this element for the inference of local, two-dimensional, steady-state flow fields. We provide a modular Python code coupling a simple AEM implementation to a Markov Chain Monte Carlo (MCMC) routine, intended for preliminary explorations of plausible flow fields during model conceptualization, or simple Bayesian flow field inference in data-scarce environments.

2. Theory

In this section, we will outline the basic concepts of AEM and some of the most common elements (Section 2.1). The derivations summarized here are mainly based on and explored in much greater detail in the seminal works of Otto Strack (1989, 2017), but include minor adaptations. For our Python implementation, we largely follow the object-oriented procedure suggested by Bakker and Kelson (2009). We present the uncertainty estimation algorithm used in this study in Section 2.3. Variables in **bold** font denote vectors, matrices, or vector- or matrix-valued functions, while standard font is reserved for scalar-valued variables and functions. The equations and derivatives for all standard elements used in this study are reported in Appendix 1 and 2 (supporting information).

2.1. Analytic Element Method

As opposed to conventional numerical models, the AEM does not seek a solution in terms of hydraulic head, but instead computes a complex-valued potential for two-dimensional steady-state flow:

$$\Omega = \Phi + i\Psi \quad (1)$$

where Φ is the *discharge* (or *hydraulic*) *potential*, Ψ is the *stream function*, which is constant along each streamline, and $i = \sqrt{-1}$ is the imaginary unit. At the heart of the method lies the superposition of solutions to simpler linear differential equations—the eponymous analytic elements—to yield a more complex result:

$$\Omega = \sum_{e=1}^E f_e(z; \theta_e) \quad (2)$$

where E is the number of analytic elements, and f_e is the function for a specific analytic element parameterized by θ_e and evaluated at z , a vector of complex-valued coordinates:

$$z = x + iy. \quad (3)$$

where x and y are two coordinate components. The conversion of hydraulic potential Φ into hydraulic heads ϕ depends on the aquifer type (Equations 8.12 and 8.13, Strack, 1989):

$$\phi = \frac{\Phi + \frac{1}{2}kH^2}{kH} \quad \left(\text{confined, } \Phi \geq \frac{1}{2}kH^2 \right) \quad (4)$$

$$\phi = \sqrt{\frac{2\Phi}{k}} \quad \left(\text{unconfined, } \Phi < \frac{1}{2}kH^2 \right) \quad (5)$$

where k is the hydraulic conductivity, H is the thickness of the aquifer, and potentials and heads are defined relative to a horizontal aquifer base of zero elevation. Its inverse is:

$$\Phi = kH\phi - \frac{1}{2}kH^2 \quad \left(\text{confined, } \phi \geq H \right) \quad (6)$$

$$\Phi = \frac{1}{2}k\phi^2 \quad \left(\text{unconfined, } \phi < H \right) \quad (7)$$

Analytic elements can be broadly classified into two groups. Some elements like extraction wells, line sinks, or area sinks only have to induce a *relative* change, for example extracting a certain net amount of water. As such, they can simply be added to the stack and induce the desired flow response. We will refer to such elements as *relative elements*. Other elements, like prescribed head boundary conditions, no-flow boundaries, or inhomogeneities must enforce an *absolute* condition at certain locations. These elements must adapt their strength to the influence of other elements to fulfill their function. We will refer to such elements as *absolute elements*. The strength values of absolute elements can be found by solving a system of linear equations.

For this study, we have created a toolbox which implements a number of standard analytical elements: uniform base flow, extraction or injection wells, linear line doublets for polygonal inhomogeneities of hydraulic conductivity and no-flow boundaries, prescribed head boundaries, and area sources and sinks with uniform infiltration (Figure 1). For readers unfamiliar with AEM, the equations for these elements and the linear solver are briefly summarized and referenced in the supporting information (Appendix 1 and 2), and described in much greater detail in the works of Otto Strack (Strack 1989, 2017). Some elements, such as extraction wells or area sinks, require a farfield reference potential. In this study, we supplemented this potential with a constant, setting their potential influence at a specified distance to zero, as we are not making use of a farfield. We expand this list of features with a novel element based on a Möbius transformation, capable inducing curving, diverging and converging regional base flow without the need for pragmatic absolute boundary conditions.

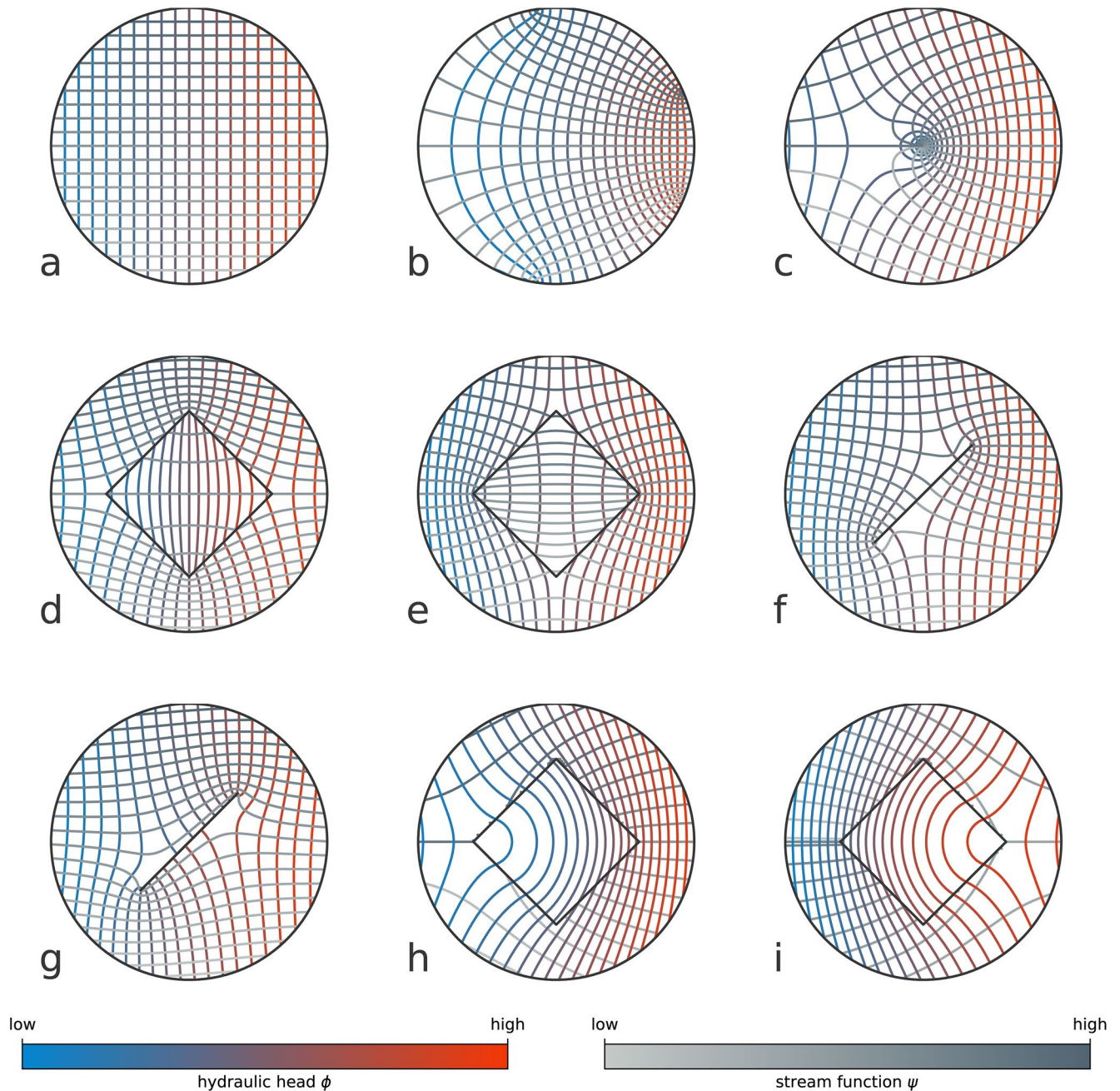


Figure 1. Various analytic elements used in this study. (a) Uniform base flow, (b) Möbius base flow, (c) rate-specified extraction well, (d) polygonal inhomogeneity (lower conductivity), (e) polygonal inhomogeneity (higher conductivity), (f) prescribed head boundary, (g) no-flow boundary, (h) extraction area sink, and (i) injection area sink. All elements are shown for confined conditions, and all elements from (c) onwards use a background of uniform flow. Solutions for the stream function are not valid inside area sinks (h and i) and have been masked inside the element. Elements which add or remove water from the system (wells, line sinks, area sinks) induce branch cuts, discontinuities in the stream function. All elements except the Möbius base flow can be embedded in infinite domains and are only shown in circular cut-outs for visual simplicity.

2.2. Möbius Base Flow

In two-dimensional flow, analytic elements can be superimposed on flow fields obtained from *conformal mapping* (e.g., Olver, 2018). Conformal mapping is a class of angle-preserving transformations which can convert grids in the complex plane—or, in our case, simple complex potential fields such as flow on the unit square from east to west—into more complex shapes, while preserving the validity of the solution. Classic

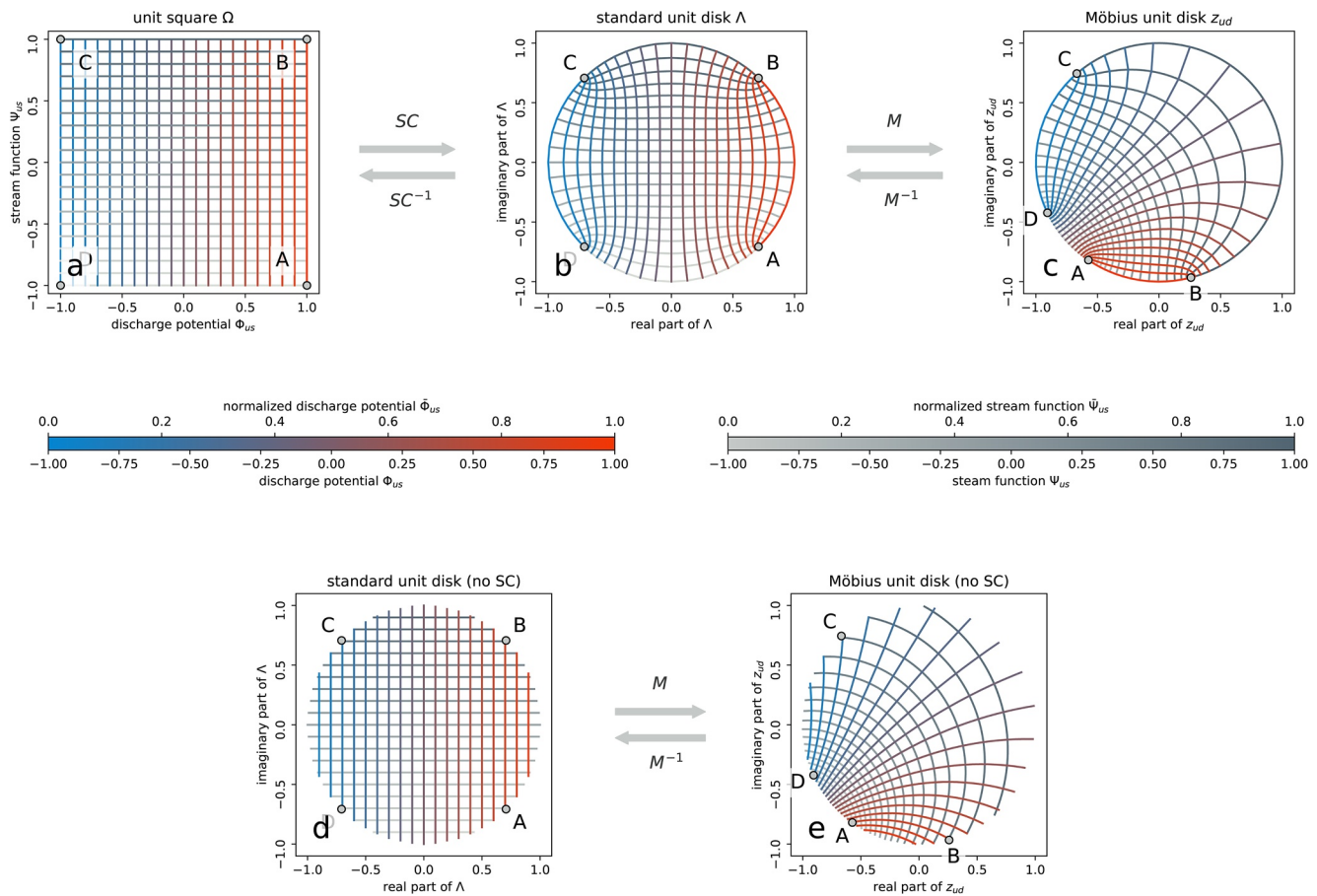


Figure 2. Illustration of the conformal mapping from the unit square (a) through the unit disk (b) to the Möbius-transformed unit disk (c). The control points A, B, and C in the unit disk (b) and the Möbius-transformed unit disk (c) define the coefficients for the Möbius transformation. Subplots (d) and (e) demonstrate the effect of a Möbius transformation on uniform flow in a circular domain without a Schwarz-Christoffel transformation, also yielding a valid solution. The forward Schwarz-Christoffel transformation is listed in Appendix 4 (supporting information).

examples of such methods are Schwarz-Christoffel (SC) transformations (e.g., Driscoll & Trefethen, 2009) and Möbius transformations (e.g., Nehari, 2012).

We can use the Möbius transformation to obtain more complex regional flow than the uniform flow analytical element can provide. The Möbius element can be set up in two variations: with or without the SC map. With the map, the SC transformation first maps the unit square onto the unit disk, then deforms it with the Möbius transformation. This process is illustrated in Figures 2a–2c. If the SC map is not used, we apply the Möbius transformation directly to uniform flow in a circular domain (Figures 2d and 2e). Both variants of the element yield slightly different flow fields. We found that this element is flexible enough to reproduce a wide range of randomized regional flows induced by farfield elements (Appendix 3 in the supporting information).

An important benefit of regional flow elements (uniform or Möbius) is that their influence on the flow field is relative. As such, their imprint on the solution is more malleable to the influence of other flow-relevant features. This reduces undesired amplifying or damping effects on the flow response of other hydrological features.

To illustrate the adverse effect of (pragmatic) absolute boundaries, consider the effect of a well with uncertain (thus variable) extraction rate near an absolute model boundary. A streamline-based no-flow boundary would reflect and amplify the drawdown response back into the model domain instead of shifting the streamline. Conversely, an interpolated prescribed head boundary condition would indiscriminately supply

the well with water from an infinite reservoir, thus damping its induced drawdown response. A malleable regional flow boundary, on the other hand, permits the well to express its drawdown response without major interference. Of course, the Möbius base flow element does not substitute (nor compromise) the use of absolute boundary conditions where appropriate: it merely renders their use optional for the generation of regional flow.

For computational purposes, we are mainly interested in the inverse of these conformal maps: from the deformed unit disk (representing points in the model domain z , Figure 2c) back onto the unit square (yielding the corresponding complex potential Ω , Figure 2a). The first step consists of casting the model domain back onto the unit disk through translation and scaling:

$$z_{ud} = \frac{z - z_d}{r_d} \quad (8)$$

where z are coordinates in the model domain, z_d is the center and r_d the radius of the circular model domain. The inverse Möbius transformation is defined as:

$$\Lambda = M^{-1}(z_{ud}; a, b, c, d) = \frac{-dz_{ud} + b}{cz_{ud} - a} \quad (9)$$

where Λ are the complex coordinates on the standard unit disk (Figure 2b), and a, b, c , and d are the Möbius coefficients. These coefficients can be computed by defining three reference points on the standard unit circle (e.g., points A, B, and C in Figure 2b) and their *images* or destinations on the edge of the transformed Möbius unit disk (i.e., A, B, and C in Figure 2c).

Why do three points suffice to define a Möbius transformation with four coefficients? To gain some intuition, consider the forward Möbius transformation:

$$z_{ud} = M(\Lambda; a, b, c, d) = \frac{a\Lambda + b}{c\Lambda + d} \quad (10)$$

With some trivial reformulation, it is possible to express this equation as:

$$z_{ud} = M\left(\Lambda; a, b, c, d\right) = \frac{a}{c} \frac{\Lambda + \frac{b}{a}}{\Lambda + \frac{d}{c}} \quad (11)$$

where we have reduced the four Möbius coefficients a, b, c , and d to three effective coefficients $a/c, b/a$, and d/c . By convention, however, Möbius transformation are often expressed in terms of the four coefficients (e.g., Nehari, 2012). To gain visual intuition about the nature of Möbius transformations, we recommend the video *Möbius Transformations revealed* by Arnold and Rogness (2008).

In search of the Möbius coefficients, it is easier to express the complex coordinates of the control points and their images in terms of polar coordinates of unit length and angle φ :

$$Z_\varphi = \cos \varphi + i \sin \varphi = e^{i\varphi} \quad (12)$$

If we consider the static reference points in Figure 2b

$$\begin{aligned} A &:= \lambda_A = e^{-i0.25\pi} \\ B &:= \lambda_B = e^{i0.25\pi} \\ C &:= \lambda_C = e^{i0.75\pi} \end{aligned} \quad (13)$$

we can express their images on the Möbius unit circle of Figure 2c similarly

$$\begin{aligned} z_A &= e^{i\varphi_A} \\ z_B &= e^{i\varphi_B} \\ z_C &= e^{i\varphi_C} \end{aligned} \quad (14)$$

Consequently, we can specify the Möbius transformation through three variables only: φ_A , φ_B , and φ_C . We can then calculate the desired Möbius coefficients (see Appendix 5 for the derivation, supporting information):

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} z_B - z_C & z_A z_C - z_A z_B \\ z_B - z_A & z_A z_C - z_C z_B \end{bmatrix}^{-1} \cdot \begin{bmatrix} \lambda_B - \lambda_C & \lambda_A \lambda_C - \lambda_A \lambda_B \\ \lambda_B - \lambda_A & \lambda_A \lambda_C - \lambda_C \lambda_B \end{bmatrix} \quad (15)$$

If desired, we may further map the standard unit disk onto the unit square with an inverse SC transformation (Fong, 2019)

$$\Omega_{us} = SC^{-1}(\Lambda) = \frac{1-i}{-K_e} F\left(c = \cos^{-1}\left(\frac{1+i}{\sqrt{2}} \Lambda\right), m = \frac{1}{\sqrt{2}}\right) + 1 - i \quad (16)$$

where Ω_{us} is the complex potential on the unit square, $F(c, m)$ is the incomplete Legendre elliptical integral of the first kind with argument c and parameter m :

$$F(c, m) = \int_0^c \frac{1}{\sqrt{1 - m \sin^2 t}} dt \quad (17)$$

and $K_e = F(c = 0.5\pi, m = 0.5) \approx 1.854$. This transformation is illustrated in Figures 2a and 2b. Note that we also obtain a valid solution if we do not make use of the SC transformation (Figures 2d and 2e). Ω_{us} (or Λ , if the SC transformation was not used) may then be transformed into a user-specified range:

$$\Omega = (\Phi_{\max} - \Phi_{\min})(\Omega_{us} + 1) / 2 + \Phi_{\min} \quad (18)$$

A further example of a Möbius base flow is shown in Figure 1b. A final note pertaining to the Möbius transformation: the list of its possible operations also includes *inversion* (i.e., the mapping of the disk's inside to its outside and vice versa), which is of course undesirable for our purposes. To check whether a given Möbius transformation of the unit disk includes inversion, it suffices to verify that the origin is not mapped outside the unit disk, at which point inversion would occur. A simple application of Equation 10 reveals that inversion occurs whenever $|b/d| > 1$. The supporting information contains a video which illustrates the resulting support for combinations of φ_A , φ_B , and φ_C .

2.3. Bayesian Inference

Bayesian statistics are a formalized way of assigning probability densities, which can be interpreted as a plausibility metric, to different alternative hypotheses, often defined as a vector of unknown parameters $\theta = [\theta_1, \dots, \theta_D]^\top$, where D is the number of uncertain parameters. When the number of hypotheses is infinite, for example in the case of continuous variables, Bayesian inference works with probability densities instead of probabilities. These densities are defined through *probability density functions (pdf)*, which assign to each hypothesis (i.e., specific parameter vector) a value corresponding to its relative plausibility. Bayes' Theorem formalizes the process of updating one's state of knowledge by combining initial belief—the prior $p(\theta)$ —with new information—the likelihood $p(y|\theta)$ —to obtain the posterior pdf $p(\theta|y)$ after normalization through the model evidence $p(y)$ (e.g., Gelman et al., 2013; Kruschke, 2015; Sivia & Skilling, 2006):

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)} \quad (19)$$

The prior $p(\theta)$ is generally user-specified. In practice, the definition of priors provides an opportunity to introduce expert knowledge. As a general rule of thumb, prior distributions should be chosen so that they return high probability densities in regions of sensible parameter values (“the hydraulic conductivity of this gravel unit should be around 10^{-2} m/s”), low densities for unrealistic parameter values (“the hydraulic conductivity of this gravel unit should be around 10^{-9} m/s”), and zero for impossible parameter values (“the hydraulic conductivity of this gravel unit should be around -5 m/s”). Confident, narrow priors will bias the posterior solution toward the initial belief, so if in doubt, it is safer to use broader, more uninformative priors, which puts more weight on the likelihood—and consequently the data-fit—during the inference. Note that priors should never be defined based on the information used for the likelihood.

The likelihood $p(y | \theta)$ expresses the probability of obtaining the observations y given the hypothesis θ and is often based on the observation error. In practical terms, it constitutes a measure of fit between the model predictions and observations. Unfortunately, it is generally impossible to solve Equation 19 analytically. The model evidence is rarely known, and it is not always possible to find a tractable closed-form formulation for the posterior.

2.3.1. Markov Chain Monte Carlo

However, even if Equation 19 does not have a closed-form solution, inference methods such as MCMC can still *sample* from the unknown posterior (e.g., Kruschke, 2015). This is useful because a sufficiently large sample set from a random distribution can act as a surrogate for the distribution itself, and consequently be used to infer its properties.

MCMC achieves this by starting from an initial hypothesis—the start point θ_0 —then exploring similar hypotheses nearby by sampling from a proposal distribution, exploring the neighborhood of the current hypothesis. MCMC then assembles a chain of samples by comparing each new hypothesis (the *proposal*) with the chain's last entry (the *reference*), and accepts or rejects the proposal based on its unnormalized posterior density relative to the reference's. If the proposal density is symmetric (it is equally probable to jump from the reference to the proposal than from the proposal to the reference), the acceptance probability p_{accept} can be calculated as:

$$p_{\text{accept}} = \min \left(1, \frac{p(\theta_{\text{proposal}}) p(y | \theta_{\text{proposal}})}{p(\theta_{k-1}) p(y | \theta_{k-1})} \right) \quad (20)$$

Equation 20 states that if the proposal is more plausible than the reference, the proposal is automatically accepted ($p_{\text{accept}} = 1$) and appended to the chain ($\theta_k = \theta_{\text{proposal}}$). If it is less plausible, it is either accepted with a probability equal to the posterior density ratio ($p_{\text{accept}} < 1$) and appended to the chain ($\theta_k = \theta_{\text{proposal}}$), or otherwise rejected, in which case the reference is appended to the chain once more ($\theta_k = \theta_{k-1}$).

2.3.2. Adaptive Proposals

A practical challenge in MCMC is that if the proposal distribution is sub-optimal, the chain will either accept or reject an inordinate amount of proposals, and consequently contain either only few independent or few unique samples. To address this issue, adaptive MCMC algorithms (e.g., Andrieu & Thoms, 2008) seek to learn an efficient proposal distribution while the chain is assembled.

In our study and toolbox, we employ a version of the Differential Evolution Markov Chain (DE-MC: Ter Braak, 2006) algorithm. This algorithm runs multiple *walkers* in parallel, each assembling its own Markov Chain, and proposes jumps based on relative differences between the chains' latest entries

$$\theta_{\text{proposal}, i} = \theta_{k-1, i} + \lambda (\theta_{k-1, R_1} - \theta_{k-1, R_2}) + \epsilon \quad (21)$$

where the second subscript i denotes the indices of different walkers, R_1 and R_2 are indices drawn without replacement from the set of walkers without i , λ is a scaling factor, and $\epsilon \sim \mathcal{N}(0, \mathbf{I}\omega)$ is a small perturbation

drawn from a multivariate Gaussian with small variance ω in order to prevent confinement to a subspace. Following Strens et al. (2002), we sample λ from a stationary beta distribution $\beta(a = 1, b = 3)$.

When using proposals based on relative differences between walkers, care should be taken to ensure that the number of walkers is larger than the number of dimensions D , as the proposals would otherwise be confined to a subspace. Ter Braak (2006) recommends choosing the number of walkers as $2D$. Finally, the full chain is assembled by concatenating the individual walkers' chains. MCMC usually also requires some post-processing for finite chains. In addition to discarding a static burn-in period, we employ an automated routine which discards all entries of a walker's chain before it first crosses above a 10% margin around the logposterior mean of all chains.

We note that the choice of inference algorithm is ultimately up to the user. In this study, we chose to implement a DE-MC algorithm due to its relative simplicity and robustness, as well as to limit the accompanying toolbox' external dependencies.

3. Examples

In this section, we illustrate the performance of the algorithm for two test cases. For the first test case, we benchmark our Python AEM code against MODFLOW, a well-established numerical finite-volume method (FVM) framework. In the second test case, we demonstrate the performance of the AEM model and the MCMC inference mechanism in a synthetic test case. The codes for both scenarios are provided under <https://doi.org/10.25678/00044N>.

3.1. Benchmarking

For benchmarking, we compare the results from our Python AEM implementation to the results of a steady-state MODFLOW 6 model (Bakker et al., 2016; Langevin et al., 2017) at varying grid resolutions. Toward this end, we design a simple synthetic model employing every element described in Section 2.1 except the base flow elements (uniform or Möbius), as both are redundant in a domain enclosed by absolute boundaries, an unfortunate prerequisite of the numerical reference. A schematic illustration of the benchmarking model is illustrated in Figure 3a, and the results compared to different resolutions of uniformly sized hexagonal grids are shown in Figures 3b–3d.

The results indicate that our AEM code can accurately reproduce the FVM predictions, and that conversely the FVM results converge toward the AEM solution at finer grid sizes. This is mainly owed to the inability of rougher grid sizes to reproduce the boundaries and features in Figure 3a as faithfully as finer resolutions. We do however note that unstructured grids—which may adopt the cell size and shape locally to reproduce features more precisely—can yield solutions closer to AEM than the uniform regular grids shown here.

3.2. Synthetic Test Case

To illustrate the potential of AEM for practical groundwater field inference, we explore the algorithm in a synthetic test case. Revisiting the motivating issue of uncertain boundary dynamics, we design our synthetic site as part of a larger catchment, with limited head information and uncertain boundary conditions along most of the domain's circumference. Such scenarios abound in hydrogeological practice. The setup of the synthetic scenario and its true reference parameters are illustrated in Figure 4. The water table is assumed to be observed at an extraction well and three surrounding observation wells (Figures 4 and 5a). Toward the north-west, a river intersects the domain, implemented as a prescribed head boundary from 335 m in the South-West to 345 m in the North-East. The following model parameters are assumed uncertain:

- c_1, c_2, c_3, c_4 : The river's connectivity to the aquifer is interpolated between four nodes at 0%, 33%, 66%, and 100% of its length.
- $k_{\text{inhomogeneity}}, k_{\text{background}}$: The northern part of the model domain is defined as a polygonal inhomogeneity with a different hydraulic conductivity than the southern part.
- $\varphi_A, \varphi_B, \varphi_C$: the flow direction control points specifying the direction, curvature, and divergence for the Möbius base flow element are set as uncertain.

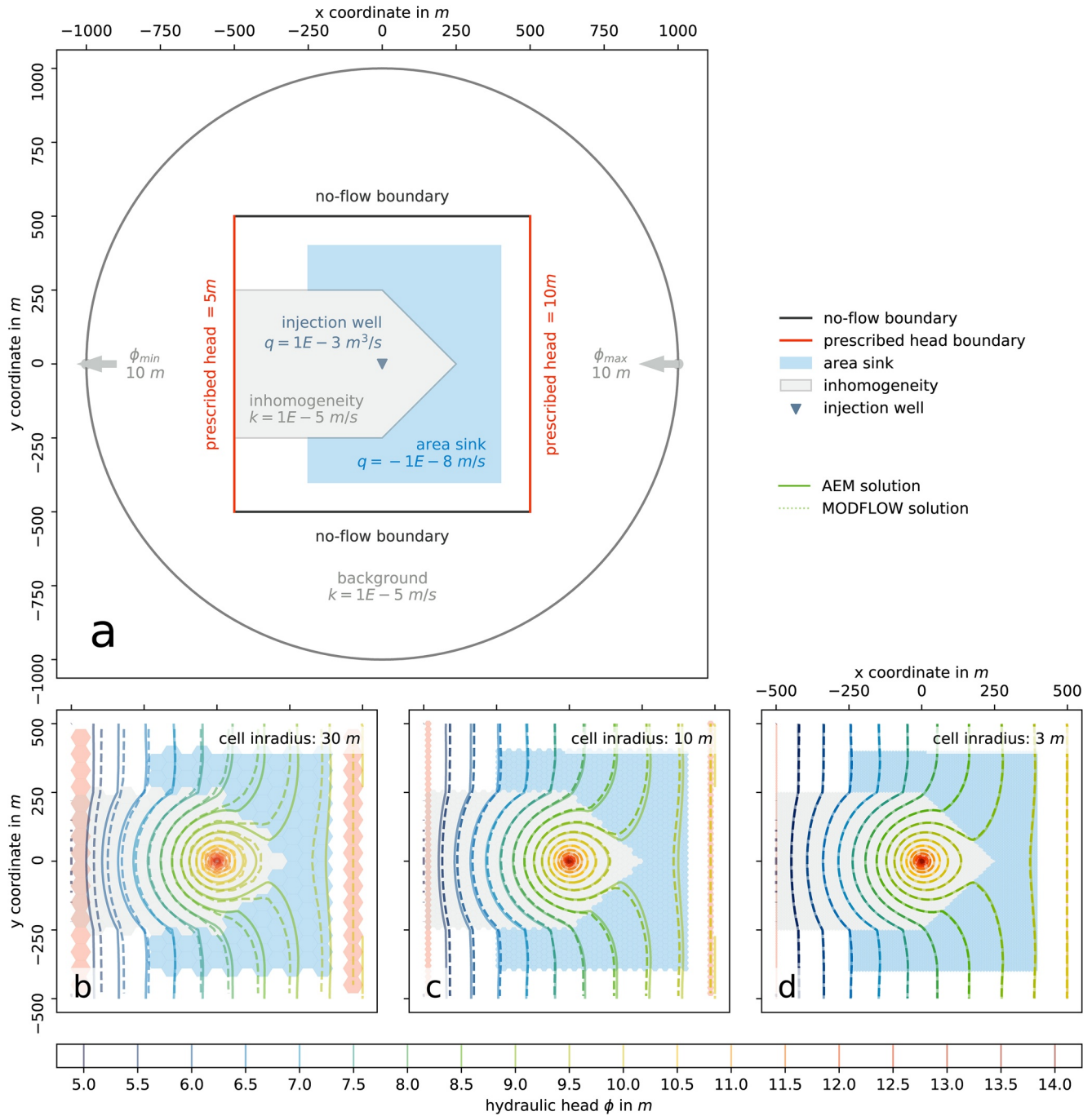


Figure 3. Illustrations of the benchmarking test case. To define a system which can be equivalently evaluated numerically, we embed a square area enclosed by two no-flow and two prescribed head boundaries within (functionally irrelevant) uniform flow. The central model domain, completely encased by absolute boundary elements, includes an area sink, an inhomogeneity, and an injection well (a). We compare the analytic element method (AEM) results (solid line; b–d) with three finite-volume method (FVM) grids of different cell in radii (radius from the hexagon's center to its edges) (dashed line): 30 m (b), 10 m (c), and 3 m (d). FVM results converge against the AEM solution at finer grid sizes. The discrepancies arise because numerical grid sizes determine how finely flow-relevant features can be resolved. Element and features outside the model domain in subplot (a) are merely shown for completeness' sake.

- h_{\min}, h_{\max} : the lower and upper hydraulic head values for the regional base flow are similarly assumed to be flexible.

In the south of the model domain, we prescribe a no-flow boundary representing an impermeable geological formation. We note that it is sometimes useful to extend flow-relevant features outside the circular

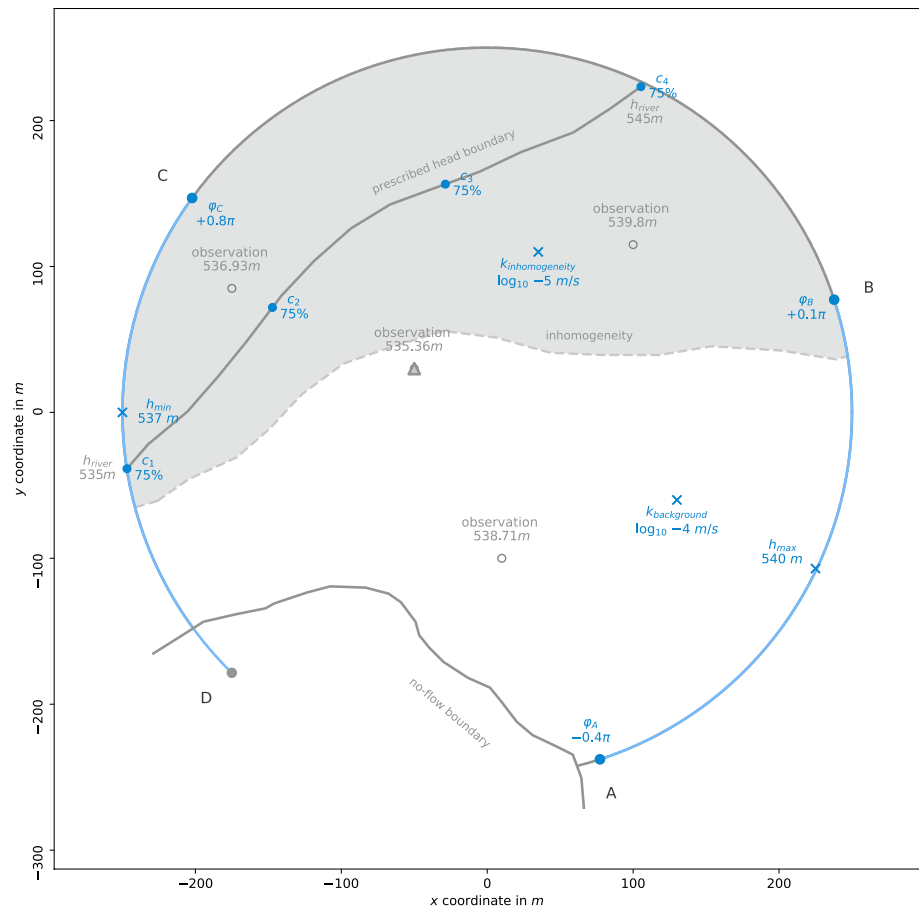


Figure 4. Synthetic test case setup, illustrating uncertain variables (blue) and fixed features (gray). The central well extracts groundwater with a rate of $-0.0025 \text{ m}^3/\text{s}$, and the river acts as a prescribed head boundary linearly interpolated between 445 m in the North-East and 435 m in the South-West. The true values for the synthetic reference are listed beneath their respective parameter labels.

model domain to prevent tip effects near the domain's edge. We extended the no-flow boundary beyond the circular model domain, but limited the river and inhomogeneity to the model domain.

For the inference, the priors we use are reported in Table S1 (supporting information) and illustrated in Figure 6. We base the likelihood on independent Gaussian observation errors ($\mu = 0\text{m}$, $\sigma = 0.15\text{m}$), and we initialize the MCMC algorithm with 32 walkers with a target chain length of 5,000 each, plus an additional burn-in period of 500 steps each.

The inference results are illustrated in Figures 5 and 6. The RMSE (0.022 m) and bias (-0.003 m) reflect the model's lack of structural error, but a glance at its uncertainty (Figure 5c) reveals greater ambiguity in the unobserved system states. Uncertainty in the groundwater field is lowest between the observation wells and swiftly increases toward the domain's edges, illustrating the diversity of different flow fields compatible with the observations. Overall, the (combined) chain's posterior predictions were faithful to the observations, with 95.73% of its predictions falling within two likelihood standard deviations of their respective observation values (Figure 5f).

To test our DE-MC implementation, we compare the inference results to the EMCEE Python module (Foreman-Mackey et al., 2013) with the same number of walkers and chain length (Figure 6, Figures S4 and S5, supporting information). Comparing MCMC chains is less trivial than it is for deterministic algorithms, but over the limited chain length we investigated, both algorithms returned similar distributions. We also note that both EMCEE and DE-MC required post-processing (manual for EMCEE, automated for DE-MC), as both returned a few walkers which never converged.

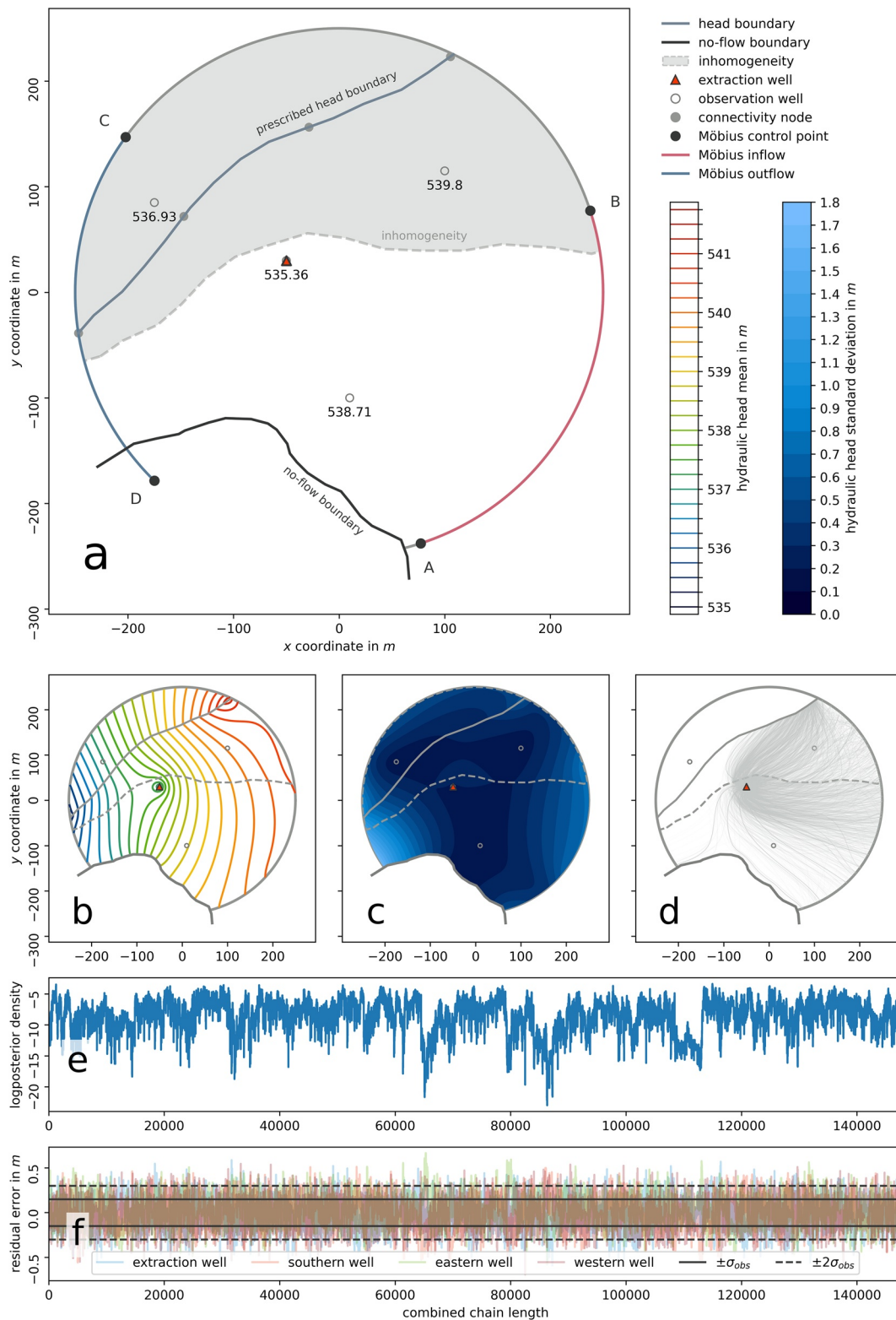


Figure 5. Schematic illustration (a) and results (b–f) for the synthetic test case. The model is implemented on a Möbius base with a polygonal inhomogeneity, an extraction well, a prescribed head boundary with spatially interpolated connectivity, and a no-flow boundary. Hydraulic heads are observed at three observation wells and the pumping well. Subplots (b and c) show the posterior mean and standard deviation of hydraulic head. Subplot (d) shows a selection of pathlines toward the extraction well. Subplot (e) shows the proportional logposterior density of the Markov Chain Monte Carlo, and subplot (f) shows the corresponding residual errors for the four observation wells. 95.73% of the chain's predictions fall within two likelihood standard deviations around the observations.

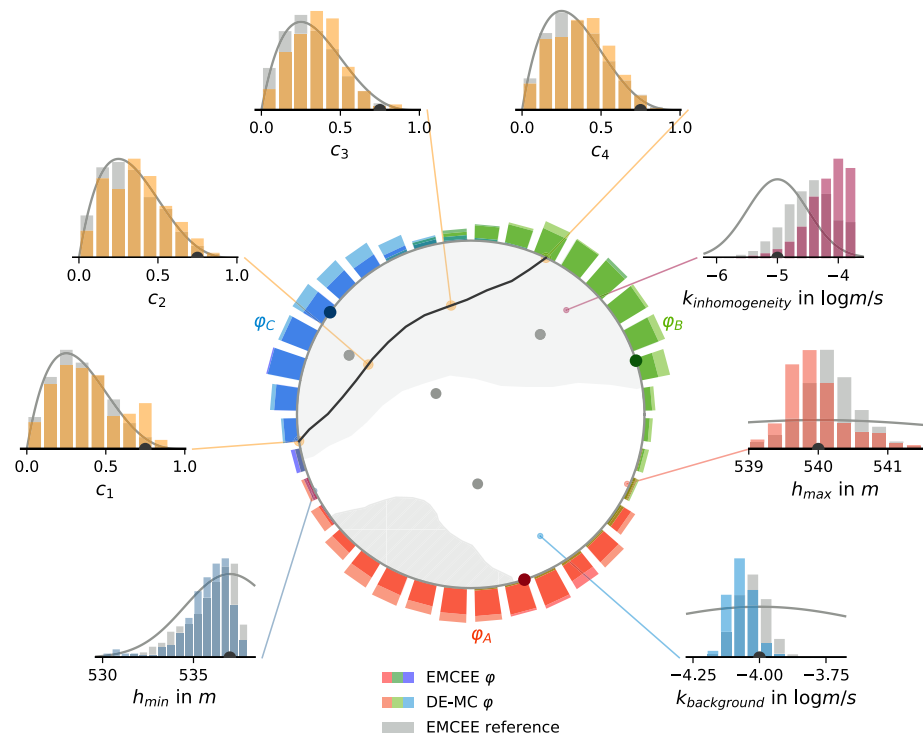


Figure 6. Comparison between parameter inference results for our Differential Evolution Markov Chain (DE-MC) implementation and results obtained using the Python EMCEE toolbox. Posterior estimates for the control point rotations are illustrated as histograms around the model domain. Gray bars are used for the EMCEE results and colors for the DE-MC results, with the exception of the Möbius rotation histograms along the circumference, for which we retained common colors to distinguish different control points. For these variables, saturation denotes the difference between the EMCEE reference (saturated) and DE-MC (desaturated). Gray lines behind the histograms denote the prior distributions, except for the Möbius control point rotations, which have an (undepicted) uniform von Mises prior. Scatter points within the histograms mark the synthetic truth.

Although this synthetic scenario is somewhat simplistic, we can illustrate some of the potential of AEM beyond simple groundwater field inference. Figure 5d illustrates a selection of uncertain flow paths toward the extraction well, which may serve as the basis for Lagrangian transport modeling. Similarly, we could investigate the sign and magnitude of the river segments' strength values to obtain probabilistic estimates about which parts of the river might be losing or gaining.

Using the algorithm as a support tool for model conceptualization, we can explore its use for the assignment of numerical model boundaries. Such a transition to numerical models might be motivated by Eulerian transport simulations or the investigation of transient dynamics. Naturally, it would be possible to calculate regional fluxes based on the AEM predictions and assign these as inflow or outflow boundaries. For transient dynamics, however, the user might wish to return to no-flow boundaries and time-variable prescribed head boundaries, which are more easily informed through marginal observation wells. If the observations and priors for the steady state AEM simulation were sufficiently representative of the average dynamics, either boundary type is best assigned in regions where the posterior AEM flow direction is relatively certain (deep blue regions, Figure 7).

The map in Figure 7 can be obtained by capitalizing on AEM's analytical nature to directly evaluate the hydraulic potential gradient $\partial\Phi / \partial z$ for each entry in the MCMC chain. The resulting set of gradients at each individual location $z \in \mathcal{Z}$ can be converted to a set of flow directions $\alpha = \arctan(\Im(\partial\Phi / \partial z) / \Re(\partial\Phi / \partial z))$, where $\Re(\cdot)$ and $\Im(\cdot)$ are the real and imaginary components of their respective arguments. Since the orientation of flow is irrelevant for the assignment of boundaries (i.e., $\alpha = -0.25\pi \triangleq 0.75\pi$), we can furthermore offset all angles $\alpha < 0$ by π (assuming $-\pi \geq \alpha > \pi$), then doubling α and halving the standard deviation

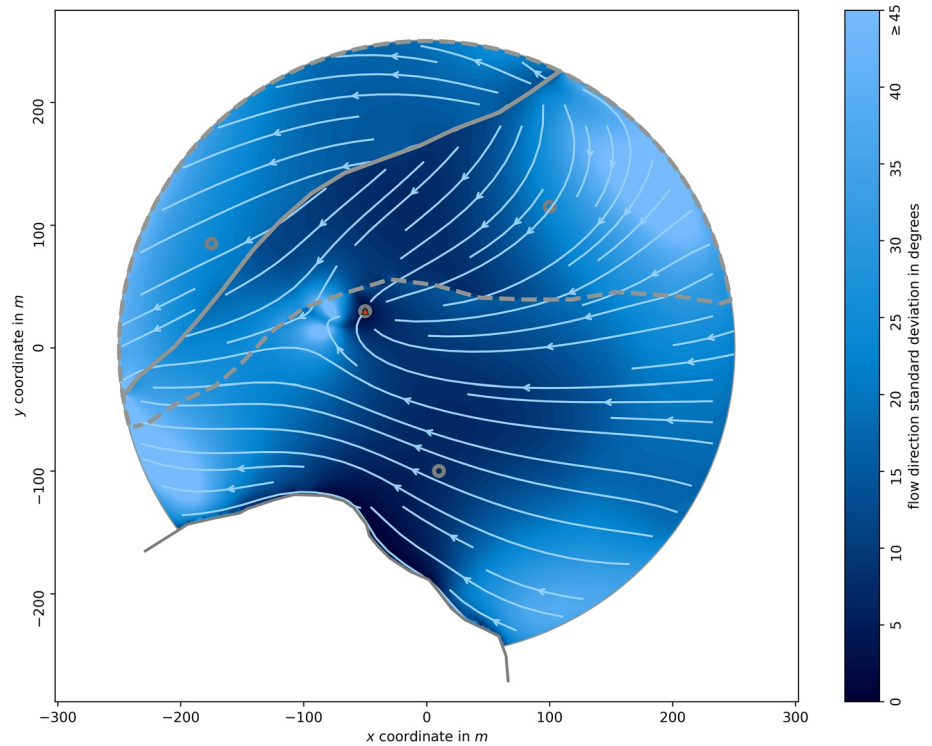


Figure 7. Posterior circular mean (streamlines) and circular standard deviation (filled contours) of the hydraulic potential gradient direction. Since the orientation (back or forth) of flow in a given direction is irrelevant for the purpose of assigning boundaries (i.e., $0.5\pi \triangleq -0.5\pi$), we estimated the circular standard deviation of the flow direction angles by first offsetting all negative angles by 180° , then doubling all angles, estimating the standard deviation, and halving this estimate. A version of this figure with simple circular standard deviation estimates is provided in Figure S6 (supporting information).

estimate to obtain a more concise estimate. Circular standard deviations for all α are defined as (e.g., Mardia & Jupp, 2008):

$$\sigma_{\text{circular}} = \sqrt{-2 \log \frac{\sqrt{\left(\sum_{i=1}^N \sin \alpha_i\right)^2 + \left(\sum_{i=1}^N \cos \alpha_i\right)^2}}{N}} \quad (22)$$

Prescribed head boundaries with constant head are best assigned perpendicular to flow lines (light blue arrows) in regions of low flow-direction uncertainty (Figure 7) and low hydraulic head uncertainty (Figure 5c). No-flow boundaries are best assigned parallel to well-informed flow lines.

4. Discussion and Conclusions

In this study, we explored the use of the AEM for the inference of uncertain groundwater tables, particularly under the lens of uncertain regional flow. Toward this end, we expanded the standard toolbox of analytical elements with an element based on conformal mapping. This new element induces curving, converging, or diverging regional flow in a circular model domain of arbitrary size. We subsequently benchmarked our toolbox against a numerical model (MODFLOW 6) for varying grid resolutions.

To examine the performance of the Möbius base flow element and toolbox in practice, we explored its application in a simple synthetic scenario. Coupling the model to an MCMC routine, we sampled from its parameter posterior and simulated the corresponding uncertain groundwater flow fields. Finally, we illustrated a few possible analyses of the inferred water tables: water table uncertainty, simple pathline tracing, or decision support for the placement of numerical model boundaries.

We find that AEM can be a highly computationally efficient tool for the exploration of uncertain flow fields in data-scarce environments. Its comparatively simple structure allows for straightforward uncertainty estimation and can make it attractive for the estimation of probabilistic flow maps, particularly in studies without a primary focus on subsurface characterization. Alternatively, it can prove valuable as a support tool in preparation for more complex numerical models, especially for the assignment of boundaries.

In summary, we believe that AEM constitutes a highly attractive compromise between simplistic, often one-dimensional analytical groundwater flow solutions (e.g., method of fragments: Harr, 2006), and the sometimes debilitating complexity of full numerical models. Its usually low parameter count and high computational efficiency renders AEM naturally well-suited for most Bayesian uncertainty estimation methods. In light of the push toward more comprehensive uncertainty analyses over the past decades, we remain confident that this property in particular warrants greater attention to AEM in the future. We have provided the AEM and MCMC toolboxes used in this study in the supporting information and on GitHub under <https://maxramgraber.github.io/Simple-AEM-Toolbox/>

Data Availability Statement

The data and algorithms to reproduce the results reported in this study have been uploaded under <https://doi.org/10.25678/00044N>.

Acknowledgments

The authors want to thank Prof. Otto Strack, University of Minnesota, and two anonymous reviewers for their diligent and detailed reviews of this manuscript. The authors also express our gratitude to Prof. Otto Strack and Prof. James Craig, University of Waterloo, who answered our questions about AEM at different points during the preparation of the manuscript. The authors are also indebted to Prof. Mark Bakker, Delft University of Technology, for clarifying our understanding of the farfield reference point during the preparation of the manuscript. 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.

References

- Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18, 343–373. <https://doi.org/10.1007/s11222-008-9110-y>
- Arnold, D. N., & Rogness, J. P. (2008). Möbius transformations revealed. *Notices of the American Mathematical Society*, 55(10), 1126–1231.
- Bakker, M. (2006). An analytic element approach for modeling polygonal inhomogeneities in multi-aquifer systems. *Advances in Water Resources*, 29, 1546–1555. <https://doi.org/10.1016/j.advwatres.2005.11.005>
- Bakker, M. (2013). Semi-analytic modeling of transient multi-layer flow with TTim. *Hydrogeology Journal*, 21, 935–943. <https://doi.org/10.1007/s10040-013-0975-2>
- Bakker, M., & Kelson, V. A. (2009). Writing analytic element programs in Python. *Ground Water*, 47, 828–834. <https://doi.org/10.1111/j.1745-6584.2009.00583.x>
- 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>
- Bakker, M., & Strack, O. D. L. (2003). Analytic elements for multiaquifer flow. *Journal of Hydrology*. [https://doi.org/10.1016/S0022-1694\(02\)00319-0](https://doi.org/10.1016/S0022-1694(02)00319-0)
- Braak, C. J. F. T., & Cajo, J. F. (2006). A Markov Chain Monte Carlo version of the genetic algorithm differential evolution: Easy Bayesian computing for real parameter spaces. *Statistics and Computing*, 16, 239–249. <https://doi.org/10.1007/s11222-006-8769-1>
- Craig, J. R. (2009). Analytic elements for flow in harmonically heterogeneous aquifers. *Water Resources Research*, 45(6). <https://doi.org/10.1029/2009WR007800>
- Craig, J. R., Janković, I., Bakker, M., & Matott, S. (2009). *Visual AEM*.
- Driscoll, T. A., & Trefethen, L. N. (2009). *Schwarz-Christoffel mapping*. Cambridge University Press. <https://doi.org/10.1017/cbo9780511546808>
- Erdal, D., & Cirpka, O. A. (2016). Joint inference of groundwater-recharge and hydraulic-conductivity fields from head data using the ensemble Kalman filter. *Hydrology and Earth System Sciences*, 20, 555–569. <https://doi.org/10.5194/hess-20-555-2016>
- Fitts, C. R., Godwin, J., Feiner, K., Mclane, C., & Mullendore, S. (2015). Analytic element modeling of steady interface flow in multilayer aquifers using AnAqSim. *Groundwater*, 53, 432–439. <https://doi.org/10.1111/gwat.12225>
- Fong, C. (2019). *Analytical methods for squaring the disc*. ArXiv Preprint ArXiv:1509.06344.
- Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013). Emcee: The MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125, 306–312. <https://doi.org/10.1086/670067>
- Furman, A., & Neuman, S. P. (2003). Laplace-transform analytic element solution of transient flow in porous media. *Advances in Water Resources*, 26, 1229–1237. <https://doi.org/10.1016/j.advwatres.2003.09.003>
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Aki, V., & Rubin, D. B. (2013). *Bayesian data analysis*. CRC Press. <https://doi.org/10.1201/b16018>
- Guillaume, J. H. A., Hunt, R. J., Comunian, A., Blakers, R. S., & Fu, B. (2016). Methods for exploring uncertainty in groundwater management predictions. In A. J. Jakeman, O. Barretero, R. J. Hunt, J.-D. Rinaudo, & A. Ross (Eds.), *Integrated groundwater management: Concepts, approaches and challenges* (pp. 711–737).
- Haitjema, H. M. (1985). Modeling three-dimensional flow in confined aquifers by superposition of both two- and three-dimensional analytic functions. *Water Resources Research*, 21, 1557–1566. <https://doi.org/10.1029/WR021i010p01557>
- Haitjema, H. M. (1995). Analytic element modeling. In *Analytic element modeling of groundwater flow*.
- Harr, M. (2006). Groundwater and seepage. In *The handbook of groundwater engineering* (2nd ed.).
- Höge, M., Guthke, A., & Nowak, W. (2019). The hydrologist's guide to Bayesian model selection, averaging and combination. *Journal of Hydrology*, 572, 96–107.
- Janković, I., & Barnes, R. (1999). Three-dimensional flow through large numbers of spheroidal inhomogeneities. *Journal of Hydrology*, 226, 224–233. [https://doi.org/10.1016/S0022-1694\(99\)00141-9](https://doi.org/10.1016/S0022-1694(99)00141-9)
- Kruschke, J. (2015). *Doing Bayesian data analysis: A tutorial introduction with R JAGS, and stan*.

- Langevin, C. D., Hughes, J. D., BantaNiswonger, E. R. R. G., 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>
- 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>
- Mardia, K. V., & Jupp, P. E. (2008). *Directional statistics*.
- Nehari, Z. (2012). *Conformal mapping*. Dover Publications.
- Olver, P. J. (2018). *Complex analysis and conformal mapping* (Vol. 84).
- Peeters, L. J. M., & Turnadge, C. (2019). When to account for boundary conditions in estimating hydraulic properties from head observations? *Groundwater*, 57, 351–355.
- Renard, P. (2007). Stochastic hydrogeology: What professionals really need? *Ground Water*, 45, 531–541. <https://doi.org/10.1111/j.1745-6584.2007.00340.x>
- Sivia, D. S., & Skilling, J. (2006). *Data analysis: A Bayesian tutorial* (2nd ed.). https://doi.org/10.1142/9789812774187_0010
- Strack, O. D. L. (1989). *Groundwater mechanics*. Prentice-Hall, Inc.
- Strack, O. D. L. (2003). Theory and applications of the analytic element method. *Reviews of Geophysics*, 41. <https://doi.org/10.1029/2002RG000111>
- Strack, O. D. L. (2017). *Analytical groundwater mechanics* (1st ed.), Cambridge University Press. <https://doi.org/10.1017/9781316563144>
- Strens, M. J. A., Bernhardt, M., & Everett, N. (2002). Markov Chain Monte Carlo sampling using direct search optimization. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 602–609).
- Vrugt, J. A., ter Braak, C. J. F., Clark, M. P. J. M. H., Hyman, J. M., & Robinson, B. A. (2008). Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov Chain Monte Carlo Simulation. *Water Resources Research*, 44. <https://doi.org/10.1029/2007wr006720>