



Data assimilation with soft constraints (DASC) through a generalized iterative ensemble smoother

Xiaodong Luo¹ · William C. Cruz²

Received: 12 May 2021 / Accepted: 25 January 2022 / Published online: 9 March 2022
© The Author(s) 2022

Abstract

This work investigates an ensemble-based workflow to simultaneously handle generic, nonlinear equality and inequality constraints in reservoir data assimilation problems. The proposed workflow is built upon a recently proposed umbrella algorithm, called the generalized iterative ensemble smoother (GIES), and inherits the benefits of ensemble-based data assimilation algorithms in geoscience applications. Unlike the traditional ensemble assimilation algorithms, the proposed workflow admits cost functions beyond the form of nonlinear-least-squares, and has the potential to develop an infinite number of constrained assimilation algorithms. In the proposed workflow, we treat data assimilation with constraints as a constrained optimization problem. Instead of relying on a general-purpose numerical optimization algorithm to solve the constrained optimization problem, we derive an (approximate) closed form to iteratively update model variables, but without the need to explicitly linearize the constraint systems. The established model update formula bears similarities to that of an iterative ensemble smoother (IES). Therefore, in terms of theoretical analysis, it becomes relatively easy to transit from an ordinary IES to the proposed constrained assimilation algorithms, and in terms of practical implementation, it is also relatively straightforward to implement the proposed workflow for users who are familiar with the IES, or other conventional ensemble data assimilation algorithms like the ensemble Kalman filter (EnKF). Apart from the aforementioned features, we also develop efficient methods to handle two noticed issues that would be of practical importance for ensemble-based constrained assimilation algorithms. These issues include localization in the presence of constraints, and the (possible) high dimensionality induced by the constraint systems. We use one 2D and one 3D case studies to demonstrate the performance of the proposed workflow. In particular, the 3D example contains experiment settings close to those of real field case studies. In both case studies, the proposed workflow achieves better data assimilation performance in comparison to the choice of using an original IES algorithm. As such, the proposed workflow has the potential to further improve the efficacy of ensemble-based data assimilation in practical reservoir data assimilation problems.

Keywords History matching · Constrained data assimilation · Generalized iterative ensemble smoother · Nonlinear equality and inequality constraints · Derivative-free optimization

1 Introduction

Data assimilation aims to estimate the quantities of interest (QoI), e.g., model variables (states and/or parameters), based on certain sources of information. In a conventional setting, the sources of information largely come from

observed data, which are connected to the QoI through some underlying relations, e.g., in the forms of observation operators or forward simulators (with possible observation errors). In real-world problems, additional sources of information may also be available, e.g., in terms of the possible ranges of model variables, or certain physical laws (e.g., mass conservation) that govern the dynamical systems constituted by the model variables. These additional sources of information constitute practical constraints, and data assimilation incorporating both observed data and available constraints is often referred to as constrained data assimilation [19] within the community of data assimilation.

A constrained data assimilation problem can be tackled from different perspectives, which leads to a variety of

✉ Xiaodong Luo
xluo@norceresearch.no

¹ Norwegian Research Centre (NORCE), 5008 Bergen, Norway

² University of Stavanger, 4021 Stavanger, Norway

constrained assimilation algorithms [2, 34]. For instance, certain constrained assimilation algorithms are designed to handle a given type of constraints, e.g., equality constraints [19, 35] or inequality constraints [6, 32, 33, 40], or a mixture of both equality and inequality constraints [1, 14, 18].

There are also different ways of incorporating the constraints. For instance, when dealing with equality constraints, one may treat the equality-constraint system as a perfect observation operator without incurring any observation errors. These constraints are referred to as “perfect measurements” in [34]. Following this standpoint, one can combine the equality-constraint system and the original observation system into an augmented observation system. In this case, solving the constrained data assimilation problem boils down to simultaneously assimilate the augmented observations. Alternatively, one can adopt a two-step procedure, in which the first step is to estimate the QoI only based on the observed data, while the second step further modifies the estimated QoI to better honor the constraints, e.g., by projecting the estimated QoI obtained at the first step onto the feasible regions induced by the constraints [13, 14], or by modifying the Kalman-gain-type matrix [14], or by iteratively refining the list (active set) of model variables that violate the constraints [32]. Under this setting, it is equivalent to sequentially assimilating different sources of information into the QoI. In addition, within the framework of Bayesian data assimilation, where the QoI are probability density functions (PDFs) of model variables, one more option to account for the constraints is to impose certain restrictions on the shapes of the prior PDFs, e.g., in terms of truncated PDFs when dealing with inequality constraints [20].

When dealing with linear constraints, one can recast constrained data assimilation as a quadratic programming problem [18], which can be solved by some general-purpose numerical optimization algorithms [30]. In addition, constrained assimilation algorithms developed from different perspectives to handle linear constraints may become equivalent under suitable conditions [34]. However, when nonlinearity is present in the constraints, more sophisticated strategies are needed, similar to the situation of handling nonlinearity in unconstrained data assimilation problems. Such strategies include, e.g., the first or second order linearization of a nonlinear-constraint system [13, 15, 35, 42, 44], unscented transform [39, 41], particle approximation [43], moving horizon estimation [3], iterative approximation [9], and their combinations.

The focus of the current work is on constrained data assimilation in reservoir characterization problems. More specifically, we consider data assimilation with soft

constraints (DASC), in the sense that the constraints will be incorporated into data assimilation, but an updated reservoir model resulting from a constrained assimilation algorithm may not strictly satisfy the constraints in general [34]. The reason for considering this relaxed setting is a few fold. First of all, although in certain circumstances it might be desirable to require that the constraints be strictly honored, such a requirement is typically satisfied in assimilation problems with linear constraints, where a two-step procedure is often adopted, with the last step designed to honor the constraints.

There are two issues related to this type of two-step procedures: (1) When a correction method is applied to enforce the constraints, the optimality of the updated models obtained at the first (unconstrained) step may be lost, since there could be a conflict between the optimality criterion used at the first step and the objective of honoring the constraints at the second. For instance, if the optimality criterion is to reduce data mismatch, then by applying the second correction step, it is likely that the further modified models may have higher data mismatch than those obtained from the first step, similar to the situation in [22]; (2) When nonlinearity is present, it becomes more challenging to strictly honor the constraints. As aforementioned, a certain approximation method is often adopted to deal with the nonlinearity in the constraint system. As such, approximation errors will typically arise, making updated model variables difficult to strictly satisfy the original constraints.

Furthermore, as discussed in [34], in some cases it may be sensible to only consider soft constraints, e.g., when one does not have the precise knowledge to accurately describe the constraint system. As an example, one may consider the box (or interval) constraints imposed on, e.g., porosity of a reservoir model. A box constraint like $[0, 1]$ would be certainly correct, but may not be sharp in general. As such, it would be reasonable to adopt a narrower constraint interval, e.g., $[10^{-3}, 0.4]$, based on some prior knowledge (e.g., well log data). The selection of the lower and upper boundaries, however, is often heuristic, thus a slight violation of the box constraint may not cause a problem in practice.

An additional reason for us to consider soft constraints is its algorithmic convenience, when combining the ideas of “perfect measurements” and “multi-objective optimization”, to derive constrained data assimilation algorithms. With the idea of “perfect measurements”, we augment the constraint and the original observation systems, and assimilate all sources of information simultaneously. In the course of data assimilation, we aim to find model variables to match all sources of “data” as accurately as

possible. Nevertheless, as aforementioned, the objective of matching one particular type of “data” to a good extent may contradict that of matching another type of “data” well. In this sense, DASC is similar to a multi-objective optimization problem. In the current work, however, we do not compute the Pareto front for the sake of computational efficiency.

Our contributions in this work include the following aspects: We develop an ensemble-based workflow to simultaneously handle generic, nonlinear equality and inequality constraints based on a recently proposed umbrella algorithm, called the generalized iterative ensemble smoother (GIES) [23]. This workflow inherits the benefits of ensemble-based data assimilation algorithms, in terms of the derivative-free nature and the ability to provide uncertainty quantification to some extent. Unlike the traditional ensemble assimilation algorithms, our proposed workflow admits cost functions beyond the conventional form of nonlinear-least-squares, and in principle allows the developments of an infinite number of constrained assimilation algorithms. We recast data assimilation with constraints as a constrained optimization problem. However, instead of relying on a general-purpose numerical optimization algorithm to solve the constrained optimization problem, we derive an (approximate) closed form for model update, but without the need to explicitly linearize the constraint systems, in contrast to some recently proposed ensemble-based algorithms (e.g., [13, 15, 44]) to handle nonlinear constraints. This closed form bears similarities to the model update formula of an ordinary iterative ensemble smoother (IES) [8, 11, 25] applied to unconstrained data assimilation problems. As such, it becomes relatively easy to transit from an ordinary IES to the proposed constrained assimilation algorithms, and the implementation of the proposed algorithms also becomes more straightforward for those who are familiar with ensemble based assimilation algorithms. Apart from the aforementioned features, we also develop efficient methods to handle two noticed issues that would be of practical importance for ensemble-based constrained assimilation algorithms. These issues include the implementation of localization in the presence of constraints, and the (possible) high dimensionality induced by the constraint system.

In other words, our proposed DASC workflow (through GIES) contains the following four features: (1) The ability to simultaneously handle generic, nonlinear equality and inequality constraints; (2) No need for explicitly evaluating the gradients of the constraint systems with respect to the QoI; (3) The ability to handle high-dimensional constraint systems; (4) A tailored adaptive localization scheme to tackle the adverse effects of a small ensemble size, while taking into account the presence of constraint systems. These features can be considered as desirable practical

benefits that help promote the applicability of the proposed DASC workflow to generic constrained data assimilation problems. To the best of our knowledge, in the literature, there do not seem to exist ensemble-based, constrained data assimilation algorithms which simultaneously possess the above-mentioned features.

We note that, it is not our intention to show the proposed DASC workflow outperforms all existing, ensemble-based constrained data assimilation algorithms. Indeed, when the gradients of the constraint systems are available, then one should expect that a constrained assimilation algorithm exploiting the knowledge of these gradients perform better than the DASC workflow, if the latter only uses less accurate, ensemble-based approximations of these gradients.¹ Instead, the main strength of the proposed DASC workflow is its wider applicability to generic constrained data assimilation problems, to some of which the existing, ensemble-based constrained data assimilation algorithms may not be applicable, due to the potential challenges in these problems, in terms of, e.g., the type(s) of the constraints, the availability of the gradients of the constraint systems, or the dimensionality of the constraint systems, etc. As an example, the first case study to be presented later considers equality constraints that involve the histograms of reservoir models. In that case, it appears challenging to evaluate the gradients of a histogram with respect to reservoir model variables. As such, most (if not all) of the existing, ensemble-based constrained data assimilation algorithms cannot be directly applied to handle such equality constraints, yet the DASC workflow works well to deal with this problem, as will be shown later.

This work is organized as follows. We start from introducing the basics of the GIES, and then proceed to explain how the GIES algorithm can be employed to handle DASC problems. Next, we illustrate the performance of the proposed constrained assimilation algorithm through one 2D and one 3D case studies. Of particular interest here is the 3D case study, which contains an inequality-constraint system whose dimension is twice that of the reservoir model (in the order of 10^5). We show that with the efficient methods to handle the issues of localization in the presence of constraints and high-dimensional data induced by the constraints, the proposed constrained assimilation algorithm works well in this particular case study. This observation also implies that the proposed algorithm has the potential to handle real-field reservoir characterization problems with practical constraints. We conclude the whole work with some technical discussions and future research directions.

¹If the gradients of the constraint systems are available, they can also be exploited by the DASC workflow. For a hint, readers are referred to Appendix A of [23].

2 Generalized iterative ensemble smoother (GIES) for data assimilation with soft constraints (DASC)

2.1 Generalized iterative ensemble smoother (GIES)

Suppose that we are given a noisy observation system

$$\mathbf{d}^o = \mathbf{g}(\mathbf{m}^{tr}) + \boldsymbol{\epsilon}. \quad (1)$$

Here, $\mathbf{g} : \mathbb{D}_m \rightarrow \mathbb{D}_d$ represents a numerical forward simulator that maps an m -dimensional vector (called model vector hereafter) $\mathbf{m}^{tr} \in \mathbb{D}_m \subseteq \mathbb{R}^m$ to a d -dimensional element in the observation space $\mathbb{D}_d \subseteq \mathbb{R}^d$. The term $\boldsymbol{\epsilon}$ stands for potential contamination noise that may be present in the course of acquiring the observations \mathbf{d}^o . Without loss of generality, we assume throughout this work that the simulator \mathbf{g} is perfect.

Under the above setting, data assimilation (often as an ill-posed inverse problem) aims to find, through a certain inversion algorithm, model variables that can match the observations \mathbf{d}^o reasonably well, which is a notion close to the idea of ensemble Kalman inversion for inverse problems [16]. Prior information available for use in the inversion algorithm includes the numerical simulator \mathbf{g} , some initial guess of \mathbf{m}^{tr} , and (often assumed) statistical distribution of the noise term $\boldsymbol{\epsilon}$. In practical data assimilation problems, constraints may also be available as additional sources of prior information. Throughout this work, we assume that $\boldsymbol{\epsilon}$ follows a Gaussian distribution $N(\mathbf{0}, \mathbf{C}_d)$, with zero mean and covariance \mathbf{C}_d .

In reservoir engineering, a class of ensemble-based algorithms, called iterative ensemble smoothers (IES), e.g., [8, 11, 12, 25], are often adopted to tackle reservoir data assimilation (also known as history matching) problems. In their practical implementations, most of the IES algorithms can be related to a certain cost function in the form of nonlinear-least-squares (NLS). For instance, [25] derived an IES algorithm by finding, at each iteration step, an ensemble of N_e reservoir models that approximately minimizes the average of an ensemble of NLS-type cost functions. Concretely, suppose that at the i th iteration step, there already exists an ensemble of model variables $\mathcal{M}^i \equiv \{\mathbf{m}_j^i\}_{j=1}^{N_e}$, where j is the index of ensemble member; and N_e denotes the ensemble size. Then the IES algorithm in [25] aims to find a new ensemble of model variables $\mathcal{M}^{i+1} \equiv \{\mathbf{m}_j^{i+1}\}_{j=1}^{N_e}$ at the $(i+1)$ th iteration step, by solving the following minimum-average-cost (MAC) problem:

$$\arg \min_{\{\mathbf{m}_j^{i+1}\}_{j=1}^{N_e}} \frac{1}{N_e} \sum_{j=1}^{N_e} L_j(\mathbf{m}_j^{i+1} | \mathbf{d}_j, \mathcal{M}^i), \quad (2)$$

$$L_j(\mathbf{m}_j^{i+1} | \mathbf{d}_j, \mathcal{M}^i) \equiv \frac{1}{2} \left\{ \left(\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}) \right)^T \mathbf{C}_d^{-1} \right.$$

$$\left. \times \left(\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}) \right) + \gamma^i \left(\mathbf{m}_j^{i+1} - \mathbf{m}_j^i \right)^T \left(\mathbf{C}_m^i \right)^{-1} \left(\mathbf{m}_j^{i+1} - \mathbf{m}_j^i \right) \right\}. \quad (3)$$

The NLS-type cost function $L_j(\mathbf{m}_j^{i+1} | \mathbf{d}_j, \mathcal{M}^i)$ is defined in Eq. 3, where \mathbf{d}_j is a perturbation of \mathbf{d}^o , drawn as a sample from the Gaussian distribution $N(\mathbf{d}^o, \mathbf{C}_d)$; $\mathbf{C}_m^i = \mathbf{S}_{\mathcal{I}}^i \left(\mathbf{S}_{\mathcal{I}}^i \right)^T$ is the sample covariance matrix induced by the ensemble \mathcal{M}^i at the i th iteration step, where $\mathbf{S}_{\mathcal{I}}^i$ is a square-root matrix defined in Eq. 7 later; and γ^i is a positive coefficient that changes over the iteration step, following certain preset rules [25]. Note that in the course of formulating the MAC problem, we follow the analysis scheme in [4] and choose to perturb the observations. One may also adopt a different setting similar to that in a certain ensemble square root filter and avoids perturbing the observations, which, however, is not a topic to be investigated in the current work.

The cost function $L_j(\mathbf{m}_j^{i+1} | \mathbf{d}_j, \mathcal{M}^i)$ consists of two parts: $\frac{1}{2} \left(\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}) \right)^T \mathbf{C}_d^{-1} \left(\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}) \right)$ represents the data mismatch term that computes a weighted euclidean (ℓ_2) distance between the j th perturbed data \mathbf{d}_j and the corresponding simulated data $\mathbf{g}(\mathbf{m}_j^{i+1})$; and $\frac{1}{2} \left(\mathbf{m}_j^{i+1} - \mathbf{m}_j^i \right)^T \left(\mathbf{C}_m^i \right)^{-1} \left(\mathbf{m}_j^{i+1} - \mathbf{m}_j^i \right)$ is a regularization term, which calculates a weighted euclidean (ℓ_2) distance between a model vector \mathbf{m}_j^i at the i th iteration step and an updated one \mathbf{m}_j^{i+1} at the next iteration step. Under this choice, the effect of this regularization term is to penalize a large deviation of \mathbf{m}_j^{i+1} from \mathbf{m}_j^i under a weighted ℓ_2 norm.

The MAC problem in Eq. 2 can be approximately solved through the following IES model update formula [23, 25]:

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \mathbf{K}^i \left(\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^i) \right), \quad j = 1, 2, \dots, N_e; \quad (4)$$

$$\begin{aligned} \mathbf{K}^i &\equiv \mathbf{S}_{\mathcal{I}}^i \left(\mathbf{S}_{\mathbf{g}}^i \right)^T \left(\mathbf{S}_{\mathbf{g}}^i \left(\mathbf{S}_{\mathbf{g}}^i \right)^T + \gamma^i \mathbf{C}_d \right)^{-1} \\ &= \mathbf{S}_{\mathcal{I}}^i \left[\left(\mathbf{S}_{\mathbf{g}}^i \right)^T \mathbf{C}_d^{-1} \mathbf{S}_{\mathbf{g}}^i + \gamma^i \mathbf{I}_{N_e} \right]^{-1} \left(\mathbf{S}_{\mathbf{g}}^i \right)^T \mathbf{C}_d^{-1}; \end{aligned} \quad (5)$$

$$\bar{\mathbf{m}}^i \equiv \frac{1}{N_e} \sum_{j=1}^{N_e} \mathbf{m}_j^i; \quad (6)$$

$$\mathbf{S}_{\mathcal{I}}^i \equiv \frac{1}{\sqrt{N_e - 1}} \left[\mathbf{m}_1^i - \bar{\mathbf{m}}^i, \mathbf{m}_2^i - \bar{\mathbf{m}}^i, \dots, \mathbf{m}_{N_e}^i - \bar{\mathbf{m}}^i \right]; \quad (7)$$

$$\begin{aligned} \mathbf{S}_{\mathbf{g}}^i &\equiv \frac{1}{\sqrt{N_e - 1}} \left[\mathbf{g}(\mathbf{m}_1^i) - \mathbf{g}(\bar{\mathbf{m}}^i), \mathbf{g}(\mathbf{m}_2^i) - \mathbf{g}(\bar{\mathbf{m}}^i), \right. \\ &\quad \left. \dots, \mathbf{g}(\mathbf{m}_{N_e}^i) - \mathbf{g}(\bar{\mathbf{m}}^i) \right], \end{aligned} \quad (8)$$

where \mathbf{I}_{N_e} is a $N_e \times N_e$ identity matrix. Note that Eq. 5 contains two equivalent ways of expressing the Kalman-gain matrix \mathbf{K}^i . The first expression is the one often used in the literature, whereas the second expression can be derived by applying a certain matrix identity to the first one [23].

The reason for us to include the second expression in Eq. 5 is that later on, it will become easier for us to see the similarities and differences between the original IES and GIES algorithms.

To facilitate the discussion later, we define a generic square-root matrix

$$\mathbf{S}_{\mathcal{O}}^i \equiv \frac{1}{\sqrt{N_e - 1}} \left[\mathcal{O}(\mathbf{m}_1^i) - \mathcal{O}(\bar{\mathbf{m}}^i), \mathcal{O}(\mathbf{m}_2^i) - \mathcal{O}(\bar{\mathbf{m}}^i), \dots, \mathcal{O}(\mathbf{m}_{N_e}^i) - \mathcal{O}(\bar{\mathbf{m}}^i) \right], \tag{9}$$

where \mathcal{O} represents a certain operator that maps a model vector into another domain. The matrix $\mathbf{S}_{\mathcal{O}}^i$ is a unified representation of the square-root matrices in Eqs. 7 and 8. Indeed, for $\mathbf{S}_{\mathcal{I}}^i$ in Eq. 7, \mathcal{O} is reduced to the identity operator \mathcal{I} , such that $\mathcal{I}(\mathbf{m}) = \mathbf{m}$ for a model vector \mathbf{m} . Similarly, for $\mathbf{S}_{\mathbf{g}}^i$ in Eq. 8, the operator \mathcal{O} equals the forward reservoir simulator \mathbf{g} .

The formulation of the MAC problem in Eqs. 2–3 does not take into account constraints of model variables. As a result, the corresponding model update formulae, Eqs. 4–8, will not be able to deal with the constraints. This deficiency exists in most of the IES algorithms in the literature. As the focus of this work, we will explain later how the constraints can be naturally handled by our proposed umbrella algorithm, the generalized iterative ensemble smoother (GIES), after introducing its basics.

In a recent work, [23] introduces a class of GIES algorithms. The motivation of [23] is to extend the original IES algorithm in [25] in such a way that the underlying cost function $L_j(\mathbf{m}_j^{i+1}|\mathbf{d}_j, \mathcal{M}^i)$ goes beyond the form of NLS in general. To this end, [23] proposes to derive GIES algorithms by solving the following generalized minimum-average-cost (GMAC) problem:

$$\arg \min_{\{\mathbf{m}_j^{i+1}\}_{j=1}^{N_e}} \frac{1}{N_e} \sum_{j=1}^{N_e} L_j(\mathbf{m}_j^{i+1}|\mathbf{d}_j, \mathcal{M}^i), \tag{10}$$

$$L_j(\mathbf{m}_j^{i+1}|\mathbf{d}_j, \mathcal{M}^i) \equiv \mathcal{D} \left[\mathcal{T}(\mathbf{d}_j) - \mathcal{T}(\mathbf{g}(\mathbf{m}_j^{i+1})) \right] + \gamma_j^i \mathcal{R} \left[\mathcal{S}(\mathbf{m}_j^{i+1}) - \mathcal{S}(\mathbf{m}_j^i) \right]. \tag{11}$$

In Eq. 11, the cost function $L_j(\mathbf{m}_j^{i+1}|\mathbf{d}_j, \mathcal{M}^i)$ also consists of two terms, namely, a data mismatch term with respect to the distance metric $\mathcal{D} : \mathbb{D}_t \subseteq \mathbb{R}^t \rightarrow [0, +\infty)$, and a regularization term with respect to the regularization operator $\mathcal{R} : \mathbb{D}_s \subseteq \mathbb{R}^s \rightarrow [0, +\infty)$. Within the data mismatch term, a transform operator $\mathcal{T} : \mathbb{D}_d \rightarrow \mathbb{D}_t$ is introduced. The presence of \mathcal{T} aims to accommodate the possibility that the observed and simulated data may be first transformed into another domain (e.g., [26, 36]) before calculating their discrepancies. In a similar way, within the regularization term, a transform operator $\mathcal{S} : \mathbb{D}_m \rightarrow \mathbb{D}_s$ is adopted to take into account the possibility that

the regularization operator may be applied to transformed model variables (e.g., [5, 17, 21]). In addition, as will be illustrated in the case studies later, the choice of the regularization parameter γ_j^i in Eq. 11 may depend on both the iteration step and the ensemble member. This is different from the situation in Eq. 3, where the regularization parameter γ^i only depends on the iteration step.

It is shown in [23] that the GMAC problem defined by Eqs. 10 and 11 is approximately solved by the following model update formula (referred to as the GIES algorithm):

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \tilde{\mathbf{K}}_j^i \nabla_{\mathcal{D}} \left[\mathcal{T}(\mathbf{d}_j) - \mathcal{T} \circ \mathbf{g}(\mathbf{m}_j^i) \right]; \tag{12}$$

$$\tilde{\mathbf{K}}_j^i \equiv \mathbf{S}_{\mathcal{I}}^i \left[\mathbf{M}_{\mathcal{D}}^i(\mathbf{d}_j) + \gamma^i \mathbf{M}_{\mathcal{R}}^i(\mathbf{m}_j^i) \right]^{-1} \left(\mathbf{S}_{\mathcal{T} \circ \mathbf{g}}^i \right)^T; \tag{13}$$

$$\mathbf{M}_{\mathcal{D}}^i(\mathbf{d}_j) \equiv \left(\mathbf{S}_{\mathcal{T} \circ \mathbf{g}}^i \right)^T \nabla_{\mathcal{D}}^2 \left[\mathcal{T}(\mathbf{d}_j) - \mathcal{T} \circ \mathbf{g}(\bar{\mathbf{m}}^i) \right] \mathbf{S}_{\mathcal{T} \circ \mathbf{g}}^i; \tag{14}$$

$$\mathbf{M}_{\mathcal{R}}^i(\mathbf{m}_j^i) \equiv \left(\mathbf{S}_{\mathcal{S}}^i \right)^T \nabla_{\mathcal{R}}^2 \left[\mathcal{S}(\bar{\mathbf{m}}^i) - \mathcal{S}(\mathbf{m}_j^i) \right] \mathbf{S}_{\mathcal{S}}^i, \tag{15}$$

where, given an operator \mathcal{O} and a dummy vector variable \mathbf{x} (and a concrete value of \mathbf{x} at \mathbf{x}_0),

$$\begin{aligned} \nabla_{\mathcal{O}}[\mathbf{x}_0] &\equiv \left. \frac{\partial \mathcal{O}}{\partial \mathbf{x}} \right|_{\mathbf{x}_0}; \\ \nabla_{\mathcal{O}}^2[\mathbf{x}_0] &\equiv \left. \frac{\partial^2 \mathcal{O}}{\partial \mathbf{x}^2} \right|_{\mathbf{x}_0}; \end{aligned} \tag{16}$$

and $\mathcal{T} \circ \mathbf{g}$ represents the composition of the transform operator \mathcal{T} and the forward simulator \mathbf{g} , such that $\mathcal{T} \circ \mathbf{g}(\mathbf{m}) \equiv \mathcal{T}(\mathbf{g}(\mathbf{m}))$ for a given model vector \mathbf{m} . Meanwhile, the definitions of $\mathbf{S}_{\mathcal{T} \circ \mathbf{g}}^i$ and $\mathbf{S}_{\mathcal{S}}^i$ follow the custom in Eq. 9.

Note that, when applying Eq. 16 to evaluate the first- and second-order derivatives in Eqs. 12–15, the operator \mathcal{O} corresponds to the distance metric \mathcal{D} or the regularization operator \mathcal{R} , and \mathbf{x} to an intermediate variable in the transformed data or model space. As such, when evaluating the derivatives like $\nabla_{\mathcal{D}} \left[\mathcal{T}(\mathbf{d}_j) - \mathcal{T} \circ \mathbf{g}(\mathbf{m}_j^i) \right]$, $\nabla_{\mathcal{D}}^2 \left[\mathcal{T}(\mathbf{d}_j) - \mathcal{T} \circ \mathbf{g}(\bar{\mathbf{m}}^i) \right]$ or $\nabla_{\mathcal{R}}^2 \left[\mathcal{S}(\bar{\mathbf{m}}^i) - \mathcal{S}(\mathbf{m}_j^i) \right]$, there is no need for further evaluating the first- or second-order gradient of $\mathcal{T} \circ \mathbf{g}$ or \mathcal{S} with respect to model variables (i.e., here one should not apply the chain rule to compute the total derivatives). Normally, since one has the freedom to choose the distance metric \mathcal{D} and the regularization operator \mathcal{R} , here we assume that \mathcal{D} and \mathcal{R} are chosen in such a way that the analytic forms of the derivatives $\nabla_{\mathcal{D}} \left[\mathcal{T}(\mathbf{d}_j) - \mathcal{T} \circ \mathbf{g}(\mathbf{m}_j^i) \right]$, $\nabla_{\mathcal{D}}^2 \left[\mathcal{T}(\mathbf{d}_j) - \mathcal{T} \circ \mathbf{g}(\bar{\mathbf{m}}^i) \right]$ or $\nabla_{\mathcal{R}}^2 \left[\mathcal{S}(\bar{\mathbf{m}}^i) - \mathcal{S}(\mathbf{m}_j^i) \right]$ are available. If this is not the case, then one can also follow [23] to find ensemble-based approximations to them.

As an example, in Eq. 11, if both the model and data transform operators, \mathcal{S} and \mathcal{T} , respectively, are identity

operators, and the distance metric \mathcal{D} and the regularization operator \mathcal{R} , respectively, satisfy

$$\mathcal{D}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{C}_d^{-1} \mathbf{x}; \quad \mathbf{x} \equiv \mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}), \quad (17)$$

$$\mathcal{R}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T (\mathbf{C}_m^i)^{-1} \mathbf{x}; \quad \mathbf{x} \equiv \mathbf{m}_j^{i+1} - \mathbf{m}_j^i, \quad (18)$$

then the cost functions in Eqs. 3 and 11 coincide. In this case, with some linear algebra, one has

$$\nabla_{\mathcal{D}}[\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1})] = \mathbf{C}_d^{-1} (\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1})); \quad (19)$$

$$\nabla_{\mathcal{D}}^2[\mathbf{d}_j - \mathbf{g}(\tilde{\mathbf{m}}^i)] = \mathbf{C}_d^{-1}; \quad (20)$$

$$\nabla_{\mathcal{R}}^2[\tilde{\mathbf{m}}^i - \mathbf{m}_j^i] = (\mathbf{C}_m^i)^{-1}. \quad (21)$$

Inserting Eqs. 19–21 into the model update formulae of GIES, Eqs. 12–15, one can see that the model update formulae are reduced to those of the original IES, Eqs. 4–8 [23].

On the other hand, $L_j(\mathbf{m}_j^{i+1}|\mathbf{d}_j, \mathcal{M}^i)$ defined in Eq. 11 has a much greater generality/flexibility than that in Eq. 3, and this forms the basis for us to derive data assimilation algorithms for DASC problems, as will be shown later. As an example, [23] considered a sub-class of cost functions in the form of

$$L_j(\mathbf{m}_j^{i+1}|\mathbf{d}_j, \mathcal{M}^i) = \frac{1}{2} (\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}))^T \mathbf{C}_d^{-1} (\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1})) + \gamma_j^i \mathcal{R}[\mathcal{S}(\mathbf{m}_j^{i+1}) - \mathcal{S}(\mathbf{m}_j^i)]; \quad (22)$$

$$\mathcal{R}[\mathcal{S}(\mathbf{m}_j^{i+1}) - \mathcal{S}(\mathbf{m}_j^i)] \equiv \frac{1}{2} \sum_{k=1}^{K_{mix}} w_k \|\mathbf{B}_k (\mathcal{S}_k(\mathbf{m}_j^{i+1}) - \mathcal{S}_k(\mathbf{m}_j^i))\|_{p_k}^{q_k}; \quad w_k \geq 0. \quad (23)$$

In Eq. 22, the data mismatch term, $\frac{1}{2} (\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}))^T \mathbf{C}_d^{-1} (\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}))$, is the same as that in the original IES of [25] (cf. Eq. 3 of this work). However, as Eq. 23 indicates, the regularization term of Eq. 22 is presented as a mixture of K_{mix} different sub-regularization-terms $w_k \|\mathbf{B}_k (\mathcal{S}_k(\mathbf{m}_j^{i+1}) - \mathcal{S}_k(\mathbf{m}_j^i))\|_{p_k}^{q_k}$, where w_k , \mathbf{B}_k and \mathcal{S}_k are mixture coefficient, weight matrix and (model) transform operator, respectively, associated with the k th sub-term; p_k and q_k are some positive real numbers, with the meaning of $\|\mathbf{x}\|_{p_k}^{q_k}$ being to raise the ℓ_{p_k} norm of a dummy vector \mathbf{x} to its q_k th power. It can be shown that the regularization term, $\frac{1}{2} (\mathbf{m}_j^{i+1} - \mathbf{m}_j^i)^T (\mathbf{C}_m^i)^{-1} (\mathbf{m}_j^{i+1} - \mathbf{m}_j^i)$, in the cost function of the original IES (also cf. Eq. 3) corresponds to the choice $K_{mix} = 1$ and $p_1 = q_1 = 2$ [23]. Meanwhile, in general p_k and q_k take positive values, and they are neither required to be any integer numbers, nor be equal to each other. As such, there could be an infinite number of

possible combinations of the pair (p_k, q_k) , and this fact already means that in principle one can derive from the cost function in Eqs. 22 and 23 an infinite number of GIES algorithms, called ℓ_p^q -GIES hereafter, whose underlying cost functions would go beyond the form of NLS in general. For demonstration, [23] investigated a group (up to 30+) of ℓ_p^q -GIES algorithms, and showed that many of them outperformed the original IES of [25] in two numerical examples. For brevity, we do not provide further information of the ℓ_p^q -GIES algorithms here, and refer readers to [23] for more details.

2.2 A GIES-Based Approach to DASC Problems

Here we consider a data assimilation problem with certain constraints, in which the observation system is the same as that in Eq. 1, whereas the constraints may consist of both equality and inequality ones. Without loss of generality, for a given model vector \mathbf{m} , let us suppose that the equality and inequality constraints, respectively, are described by the following systems:

$$\mathbf{f}_{eq}(\mathbf{m}) = \mathbf{0}; \quad (24)$$

$$\mathbf{h}_{in}(\mathbf{m}) \leq \mathbf{0}; \quad (25)$$

where both \mathbf{f}_{eq} (“*eq*” for equality) and \mathbf{h}_{in} (“*in*” for inequality) could be some nonlinear vector mappings in general. For brevity, we also refer to equality and inequality constraints as EQ-constraints and IN-constraints, respectively.

As shown in the previous subsection, both the original IES and GIES algorithms can be derived by solving certain stochastic optimization problems, but without yet taking into account possible constraints on model variables. Following this optimization-theoretic perspective, including the constraints into data assimilation can then be recast as some constrained optimization problem. Bearing this notion in mind, we propose to derive a class of constrained GIES algorithms by solving the following GMAC problem:

$$\arg \min_{\{\mathbf{m}_j^{i+1}\}_{j=1}^{N_e}} \frac{1}{N_e} \sum_{j=1}^{N_e} L_j(\mathbf{m}_j^{i+1}|\mathbf{d}_j, \mathcal{M}^i); \quad (26)$$

$$L_j(\mathbf{m}_j^{i+1}|\mathbf{d}_j, \mathcal{M}^i) \equiv \mathcal{D}_{eff}(\mathbf{m}_j^{i+1}, \mathbf{d}_j) + \gamma_j^i \mathcal{R}[\mathcal{S}(\mathbf{m}_j^{i+1}) - \mathcal{S}(\mathbf{m}_j^i)]; \quad (27)$$

$$\mathcal{D}_{eff}(\mathbf{m}_j^{i+1}, \mathbf{d}_j) \equiv \mathcal{D}_d[\mathcal{T}_d(\mathbf{d}_j) - \mathcal{T}_d(\mathbf{g}(\mathbf{m}_j^{i+1}))] + \alpha_j^i \mathcal{D}_{eq}[\mathcal{T}_{eq}(\mathbf{0}) - \mathcal{T}_{eq}(\mathbf{f}_{eq}(\mathbf{m}_j^{i+1}))] + \beta_j^i \mathcal{D}_{in}[\mathcal{T}_{in}(\mathbf{0}) - \mathcal{T}_{in}(\mathbf{h}_{in}(\mathbf{m}_j^{i+1}))]. \quad (28)$$

Equations 27 and 28 indicate that the cost function underlying a constrained GIES algorithm contains an “effective” data mismatch term and a regularization term. In contrast to the data mismatch term in the unconstrained data assimilation problem (cf Eq. 11), however, now the “effective” data mismatch term needs to incorporate the impacts of constraints on the assimilation algorithm. Therefore, in Eq. 28, the “effective” data mismatch term is in the form of a mixture of three sub-terms, corresponding to the observation and constraint systems, Eqs. 1, 24 and 25, respectively. The notations in Eq. 28 are similar to those in Eq. 11, but with some additional subscripts like “*d*” (“*d*” for data), “*eq*” and “*in*” present for distinction of different sources of information. For instance, \mathcal{D}_{index} and \mathcal{T}_{index} , $index \in \{d, eq, in\}$, stand for the distance metric and the transform operator of respective (observation or constraint) system.

Without loss of generality, we assign a mixture coefficient of 1 to the term $\mathcal{D}_d [\mathcal{T}_d(\mathbf{d}_j) - \mathcal{T}_d(\mathbf{g}(\mathbf{m}_j^{i+1}))]$,

whereas α_j^i and β_j^i are mixture coefficients of respective constraint systems. Since we are dealing with soft constraints, we do not treat α_j^i and β_j^i as Lagrange multipliers. Therefore, we do not adopt the Karush–Kuhn–Tucker (KKT) conditions [30] to choose the values of α_j^i and β_j^i . Instead, α_j^i and β_j^i are treated as scalars that specify the relative weights of individual data mismatch terms, and their values will be determined in a way adaptive to the iteration step and ensemble member, using a criterion similar to that in [23], which will be illustrated in numerical case studies later.

Similar to the way to derive the ℓ_p^q -GIES algorithms, the model update formulae, Eqs. 12–15, can be applied to derive constrained assimilation algorithms, with some changes to account for the mixture of data mismatch terms. Specifically, using the analysis in Appendix D of [23], for the model update formulae in Eqs. 12–15, the following parts will be reduced to:

$$\nabla_{\mathcal{D}} [\mathcal{T}(\mathbf{d}_j) - \mathcal{T} \circ \mathbf{g}(\mathbf{m}_j^i)] = \left[\left(\nabla_{\mathcal{D}_d} [\mathcal{T}_d(\mathbf{d}_j) - \mathcal{T}_d \circ \mathbf{g}(\mathbf{m}_j^i)] \right)^T, \alpha_j^i \left(\nabla_{\mathcal{D}_{eq}} [\mathcal{T}_{eq}(\mathbf{0}) - \mathcal{T}_{eq} \circ \mathbf{f}_{eq}(\mathbf{m}_j^i)] \right)^T, \beta_j^i \left(\nabla_{\mathcal{D}_{in}} [\mathcal{T}_{in}(\mathbf{0}) - \mathcal{T}_{in} \circ \mathbf{h}_{in}(\mathbf{m}_j^i)] \right)^T \right]^T; \tag{29}$$

$$\mathbf{S}_{\mathcal{T} \circ \mathbf{g}}^i = \left[\left(\mathbf{S}_{\mathcal{T}_d \circ \mathbf{g}}^i \right)^T, \left(\mathbf{S}_{\mathcal{T}_{eq} \circ \mathbf{f}_{eq}}^i \right)^T, \left(\mathbf{S}_{\mathcal{T}_{in} \circ \mathbf{h}_{in}}^i \right)^T \right]^T; \tag{30}$$

$$\begin{aligned} \mathbf{M}_{\mathcal{D}}^i(\mathbf{d}_j) &= \left(\mathbf{S}_{\mathcal{T}_d \circ \mathbf{g}}^i \right)^T \nabla_{\mathcal{D}_d}^2 [\mathcal{T}_d(\mathbf{d}_j) - \mathcal{T}_d \circ \mathbf{g}(\mathbf{m}_j^i)] \mathbf{S}_{\mathcal{T}_d \circ \mathbf{g}}^i \\ &+ \alpha_j^i \left(\mathbf{S}_{\mathcal{T}_{eq} \circ \mathbf{f}_{eq}}^i \right)^T \nabla_{\mathcal{D}_{eq}}^2 [\mathcal{T}_{eq}(\mathbf{0}) - \mathcal{T}_{eq} \circ \mathbf{f}_{eq}(\mathbf{m}_j^i)] \mathbf{S}_{\mathcal{T}_{eq} \circ \mathbf{f}_{eq}}^i \\ &+ \beta_j^i \left(\mathbf{S}_{\mathcal{T}_{in} \circ \mathbf{h}_{in}}^i \right)^T \nabla_{\mathcal{D}_{in}}^2 [\mathcal{T}_{in}(\mathbf{0}) - \mathcal{T}_{in} \circ \mathbf{h}_{in}(\mathbf{m}_j^i)] \mathbf{S}_{\mathcal{T}_{in} \circ \mathbf{h}_{in}}^i. \end{aligned} \tag{31}$$

Inserting Eqs. 29–31 into Eqs. 12–15, we obtain the following model update formula for DASC problems:

$$\begin{aligned} &+ \alpha_j^i \left(\mathbf{S}_{\mathcal{T}_{eq} \circ \mathbf{f}_{eq}}^i \right)^T \nabla_{\mathcal{D}_{eq}} [\mathcal{T}_{eq}(\mathbf{0}) - \mathcal{T}_{eq} \circ \mathbf{f}_{eq}(\mathbf{m}_j^i)] \\ &+ \beta_j^i \left(\mathbf{S}_{\mathcal{T}_{in} \circ \mathbf{h}_{in}}^i \right)^T \nabla_{\mathcal{D}_{in}} [\mathcal{T}_{in}(\mathbf{0}) - \mathcal{T}_{in} \circ \mathbf{h}_{in}(\mathbf{m}_j^i)]. \end{aligned} \tag{32}$$

$$\begin{aligned} \mathbf{m}_j^{i+1} &= \mathbf{m}_j + \mathbf{S}_{\mathcal{I}}^i \left[\left(\mathbf{S}_{\mathcal{T}_d \circ \mathbf{g}}^i \right)^T \nabla_{\mathcal{D}_d}^2 [\mathcal{T}_d(\mathbf{d}_j) - \mathcal{T}_d \circ \mathbf{g}(\mathbf{m}_j^i)] \right. \\ &\times \mathbf{S}_{\mathcal{T}_d \circ \mathbf{g}}^i + \alpha_j^i \left(\mathbf{S}_{\mathcal{T}_{eq} \circ \mathbf{f}_{eq}}^i \right)^T \nabla_{\mathcal{D}_{eq}}^2 [\mathcal{T}_{eq}(\mathbf{0}) \\ &- \mathcal{T}_{eq} \circ \mathbf{f}_{eq}(\mathbf{m}_j^i)] \mathbf{S}_{\mathcal{T}_{eq} \circ \mathbf{f}_{eq}}^i \\ &+ \beta_j^i \left(\mathbf{S}_{\mathcal{T}_{in} \circ \mathbf{h}_{in}}^i \right)^T \nabla_{\mathcal{D}_{in}}^2 [\mathcal{T}_{in}(\mathbf{0}) - \mathcal{T}_{in} \\ &\circ \mathbf{h}_{in}(\mathbf{m}_j^i)] \mathbf{S}_{\mathcal{T}_{in} \circ \mathbf{h}_{in}}^i + \gamma_j^i \left(\mathbf{S}_{\mathcal{S}}^i \right)^T \nabla_{\mathcal{R}}^2 \\ &\left. [\mathcal{S}(\bar{\mathbf{m}}^i) - \mathcal{S}(\mathbf{m}_j^i)] \mathbf{S}_{\mathcal{S}}^i \right]^{-1} \\ &\times \left[\left(\mathbf{S}_{\mathcal{T}_d \circ \mathbf{g}}^i \right)^T \nabla_{\mathcal{D}_d} [\mathcal{T}_d(\mathbf{d}_j) - \mathcal{T}_d \circ \mathbf{g}(\mathbf{m}_j^i)] \right. \end{aligned}$$

For reference later, we call algorithms derived from Eq. 32 constrained GIES (C-GIES). Clearly, when no constraint is considered (e.g., by setting $\alpha_j^i = \beta_j^i = 0$), the C-GIES will reduce to the GIES algorithms in Eqs. 12–15. On the other hand, the impacts of EQ and/or IN-constraints will be promoted by increasing the values of α_j^i and/or β_j^i .

In practical data assimilation problems, the distance metrics \mathcal{D}_d , \mathcal{D}_{eq} and \mathcal{D}_{in} , regularization operator \mathcal{R} , and transform operators \mathcal{T}_d , \mathcal{T}_{eq} , \mathcal{T}_{in} and \mathcal{S} in Eq. 32 are chosen by the users. This implies that in principle there could also be infinitely many C-GIES algorithms, similar to what

we have previously mentioned for the ℓ_p^q -GIES algorithms. In the current work, however, our main objective is not to show the richness of the C-GIES algorithms. Instead, we focus more on explaining how the constraints can be incorporated into data assimilation through a GIES-based framework, and demonstrating the benefits of including such constraints. Consequently, in the sequel, we will only consider a particular C-GIES algorithm which is close to the original IES algorithm in [25]. In fact, this C-GIES algorithm will recover the original IES algorithm if there is no constraint present (e.g., by setting $\alpha_j^i = \beta_j^i = 0$). Therefore, under this setting, the impacts of the constraints will become manifest by comparing the performance of this C-GIES and the original IES algorithms.

Specifically, in the chosen C-GIES algorithm, we let the distance metric \mathcal{D}_d and the regularization operator \mathcal{R} be the same as those in Eqs. 17 and 18, respectively. The choices of the distance metrics \mathcal{D}_{eq} and \mathcal{D}_{in} are worthy of more attention, and will thus be discussed separately later. Meanwhile, we let $\mathcal{T}_d, \mathcal{T}_{eq}, \mathcal{T}_{in}$ and \mathcal{S} all be identity operators. With these choices, Eq. 32 is reduced to

$$\begin{aligned} \mathbf{m}_j^{i+1} = & \mathbf{m}_j^i + \mathbf{S}_{\mathcal{I}}^i \left[\left(\mathbf{S}_{\mathbf{g}}^i \right)^T \mathbf{C}_{\mathbf{d}}^{-1} \mathbf{S}_{\mathbf{g}}^i + \alpha_j^i \left(\mathbf{S}_{\mathbf{f}_{eq}}^i \right)^T \nabla_{\mathcal{D}_{eq}}^2 \right. \\ & \left[\mathbf{0} - \mathbf{f}_{eq} \left(\mathbf{m}_j^i \right) \right] \mathbf{S}_{\mathbf{f}_{eq}}^i + \beta_j^i \left(\mathbf{S}_{\mathbf{h}_{in}}^i \right)^T \nabla_{\mathcal{D}_{in}}^2 \\ & \left[\mathbf{0} - \mathbf{h}_{in} \left(\mathbf{m}_j^i \right) \right] \mathbf{S}_{\mathbf{h}_{in}}^i + \gamma_j^i \mathbf{I}_{N_e} \right]^{-1} \\ & \times \left[\left(\mathbf{S}_{\mathbf{g}}^i \right)^T \mathbf{C}_{\mathbf{d}}^{-1} \left[\mathbf{d}_j - \mathbf{g} \left(\mathbf{m}_j^i \right) \right] + \alpha_j^i \left(\mathbf{S}_{\mathbf{f}_{eq}}^i \right)^T \right. \\ & \nabla_{\mathcal{D}_{eq}} \left[\mathbf{0} - \mathbf{f}_{eq} \left(\mathbf{m}_j^i \right) \right] + \beta_j^i \left(\mathbf{S}_{\mathbf{h}_{in}}^i \right)^T \nabla_{\mathcal{D}_{in}} \\ & \left. \left[\mathbf{0} - \mathbf{h}_{in} \left(\mathbf{m}_j^i \right) \right] \right]. \end{aligned} \quad (33)$$

Note that in Eq. 33, $\mathbf{0} - \mathbf{f}_{eq} \left(\mathbf{m}_j^i \right) = -\mathbf{f}_{eq} \left(\mathbf{m}_j^i \right)$ and $\mathbf{0} - \mathbf{h}_{in} \left(\mathbf{m}_j^i \right) = -\mathbf{h}_{in} \left(\mathbf{m}_j^i \right)$. We make the value $\mathbf{0}$ explicitly present in the update formula to indicate that $\mathbf{0}$ is treated as the perfect measurements in the constraint systems.

Regarding the choice of the distance metric \mathcal{D}_{in} for IN-constraints, essentially we utilize the concept of barrier function [6, 30]. Suppose that \mathbf{m}_j^i already satisfies the IN-constraints, then the idea of barrier function is to choose a continuous mapping \mathcal{D}_{in} in such a way that $\mathcal{D}_{in} \left[\mathbf{0} - \mathbf{h}_{in} \left(\mathbf{m}_j^{i+1} \right) \right]$ will output relatively large values as $\mathbf{h}_{in} \left(\mathbf{m}_j^{i+1} \right)$ moves closer to the boundary $\mathbf{0}$, and that $\mathcal{D}_{in} \left[\mathbf{0} - \mathbf{h}_{in} \left(\mathbf{m}_j^{i+1} \right) \right]$ will become relatively small if $\mathbf{h}_{in} \left(\mathbf{m}_j^{i+1} \right)$ is within the feasible region, but relatively far away from the boundary $\mathbf{0}$. In other words, a barrier function aims to encourage a feasible estimate \mathbf{m}_j^{i+1} so that

$\mathbf{h}_{in} \left(\mathbf{m}_j^{i+1} \right)$ moves away from the boundary, as a way to honor the IN-constraints. In the current work, we choose the following form for \mathcal{D}_{in} : for a given dummy vector $\mathbf{x} \geq \mathbf{0}$ (e.g., $\mathbf{x} = -\mathbf{h}_{in} \left(\mathbf{m}_j^i \right)$),

$$\mathcal{D}_{in}(\mathbf{x}) = -(\log(\mathbf{x} + \mathbf{a}))^T \mathbf{1}_{len(\mathbf{x})}. \quad (34)$$

In Eq. 34, $\mathbf{a} > 0$ is a constant vector in the same size as \mathbf{x} ; $\log(\bullet)$ is a vector function which applies the natural logarithmic function to each element of \bullet ; and $\mathbf{1}_{len(\mathbf{x})}$ is a vector whose elements are all equal to 1, and whose length is the same as that of the vector \mathbf{x} . As desired, when any of the elements of $\mathbf{x} + \mathbf{a}$ approaches 0, then $\mathcal{D}_{in}(\mathbf{x})$ tends to $+\infty$; On the other hand, if all elements of $\mathbf{x} + \mathbf{a}$ are relatively far away from 0, then the value of $\mathcal{D}_{in}(\mathbf{x})$ will decrease. The constant vector \mathbf{a} is introduced here for numerical stability, to prevent possible divisions by zero in Eqs. 35 and 36 later. The value of \mathbf{a} will be specified in numerical case studies later, when it comes to concrete IN-constraints.

Equation 33 will require us to evaluate $\nabla_{\mathcal{D}_{in}}[\mathbf{x}]$ and $\nabla_{\mathcal{D}_{in}}^2[\mathbf{x}]$ at $\mathbf{x} = -\mathbf{h}_{in} \left(\mathbf{m}_j^i \right)$. With some algebra, it can be shown that, for $\mathcal{D}_{in}(\mathbf{x})$ defined in Eq. 34, we have

$$\nabla_{\mathcal{D}_{in}}[\mathbf{x}] = -\mathbf{1}_{len(\mathbf{x})} ./ (\mathbf{x} + \mathbf{a}); \quad (35)$$

$$\nabla_{\mathcal{D}_{in}}^2[\mathbf{x}] = \text{diag} \left(\left(\mathbf{1}_{len(\mathbf{x})} ./ (\mathbf{x} + \mathbf{a}) \right)^{\wedge 2} \right). \quad (36)$$

In Eqs. 35 and 36, $./$ and $\wedge 2$ stand for element-wise division and square operators applied to vectors, such that $\mathbf{1}_{len(\mathbf{x})} ./ (\mathbf{x} + \mathbf{a}) = [1/(x_1 + a_1), 1/(x_2 + a_2), \dots]^T$ and $\left(\mathbf{1}_{len(\mathbf{x})} ./ (\mathbf{x} + \mathbf{a}) \right)^{\wedge 2} = [1/(x_1 + a_1)^2, 1/(x_2 + a_2)^2, \dots]^T$, where x_k, a_k ($k = 1, 2, \dots$) are elements of \mathbf{x} and \mathbf{a} , respectively. The diag operator converts a vector to a diagonal matrix, in such a way that the diagonal elements of the diagonal matrix are filled in by the elements of that vector.

For the choice of the distance metric \mathcal{D}_{eq} , we adopt a notion opposite to barrier function. Instead of propelling the constraint system $(\mathbf{h}_{in} \left(\mathbf{m}_j^{i+1} \right))$ away from the boundary $\mathbf{0}$, we want \mathcal{D}_{eq} to attract the constraint system $(\mathbf{f}_{eq} \left(\mathbf{m}_j^{i+1} \right))$ towards $\mathbf{0}$. For this reason, we call \mathcal{D}_{eq} a “channel function”. In this work, \mathcal{D}_{eq} is chosen as follows: for a given dummy vector \mathbf{x} (e.g., $\mathbf{x} = \mathbf{f}_{eq} \left(\mathbf{m}_j^i \right)$),

$$\mathcal{D}_{eq}(\mathbf{x}) = (\log(|\mathbf{x}| + \mathbf{b}))^T \mathbf{1}_{len(\mathbf{x})}; \quad (37)$$

where \mathbf{b} is also a constant vector close to zero; $|\mathbf{x}|$ and $\log(\mathbf{x})$ are element-wise operators that take the absolute value, and natural logarithm of a vector \mathbf{x} , respectively. Roughly speaking, $\mathcal{D}_{eq}(\mathbf{x})$ will become relatively large if the element magnitudes of \mathbf{x} are big, and tend to decrease when the element magnitudes of \mathbf{x} are reduced. Similar to Eqs. 35 and 36, it can be shown that, the gradient and

Hessian of \mathcal{D}_{eq} with respect to \mathbf{x} in Eq. 37 is given by

$$\nabla_{\mathcal{D}_{eq}} [\mathbf{x}] = \mathbf{1}_{len(\mathbf{x})} ./ (\mathbf{x} + \mathbf{b} \dot{\times} \text{sgn}(\mathbf{x})); \tag{38}$$

$$\nabla_{\mathcal{D}_{eq}}^2 [\mathbf{x}] = -\text{diag} \left(\left(\mathbf{1}_{len(\mathbf{x})} ./ (\mathbf{x} + \mathbf{b} \dot{\times} \text{sgn}(\mathbf{x})) \right)^{\wedge 2} \right), \tag{39}$$

where $\text{sgn}(\mathbf{x}) \equiv [\text{sgn}(x_1), \text{sgn}(x_2), \dots]^T$ returns the elements' signs of \mathbf{x} , and $\dot{\times}$ denotes element-wise (Hadamard) product, such that $\mathbf{x} \dot{\times} \mathbf{y} = [x_1 \times y_1, x_2 \times y_2, \dots]^T$, with x_k and y_k ($k = 1, 2, \dots$) being the elements of \mathbf{x} and \mathbf{y} , respectively. In numerical calculations, we follow the custom to set $\text{sgn}(\mathbf{0}) = \mathbf{0}$, which implies that in Eqs. 38 and 39, divisions by zero will occur at $\mathbf{x} = \mathbf{0}$. To mitigate this potential problem, in our implementation, we approximate the gradient and Hessian in Eqs. 38 as follows:

$$\nabla_{\mathcal{D}_{eq}} [\mathbf{x}] = \mathbf{1}_{len(\mathbf{x})} ./ (\mathbf{x} + \mathbf{b} \dot{\times} \text{sgn}(\mathbf{x}) + \epsilon \mathbf{1}_{len(\mathbf{x})}); \tag{40}$$

$$\nabla_{\mathcal{D}_{eq}}^2 [\mathbf{x}] = -\text{diag} \left(\left(\mathbf{1}_{len(\mathbf{x})} ./ (\mathbf{x} + \mathbf{b} \dot{\times} \text{sgn}(\mathbf{x}) + \epsilon \mathbf{1}_{len(\mathbf{x})}) \right)^{\wedge 2} \right), \tag{41}$$

where $\epsilon > 0$ is a (relatively small) positive scalar. Again, the values of \mathbf{b} and ϵ will be specified in numerical case studies later.

For the distance metrics \mathcal{D}_{in} and \mathcal{D}_{eq} defined by Eqs. 34 and 37, respectively, one feature is that their resulting Hessian in Eqs. 36 and 39 (or 41)) are diagonal matrices. This feature would be particularly useful if the corresponding constraint systems are of very high dimensions (which is the situation in one of our numerical case studies later). To see this point, let us take the Hessian $\nabla_{\mathcal{D}_{in}}^2 [\mathbf{x}]$ as an example. Being a diagonal matrix means that $\nabla_{\mathcal{D}_{in}}^2 [\mathbf{x}]$ can be manipulated as a sparse matrix, which would consume considerably less computer memory than a non-diagonal, full Hessian matrix. In addition, the C-GIES update formula (33) involves certain matrix products like $(\mathbf{S}_{\mathbf{h}_{in}}^i)^T \nabla_{\mathcal{D}_{in}}^2 [\mathbf{0} - \mathbf{h}_{in}(\mathbf{m}_j^i)] \mathbf{S}_{\mathbf{h}_{in}}^i$. With $\nabla_{\mathcal{D}_{in}}^2 [\mathbf{0} - \mathbf{h}_{in}(\mathbf{m}_j^i)]$ in the form of a diagonal matrix, the matrix products can be simplified,² which also implies reduced computational costs. On the other hand, it should be noted that in general, there could be many good choices for \mathcal{D}_{in} and \mathcal{D}_{eq} , while those defined by Eqs. 34 and 37 may not necessarily be the optimal ones. Investigating the optimal forms of \mathcal{D}_{in} and \mathcal{D}_{eq} is beyond the scope of the current work, and is thus not further considered hereafter.

²For instance, for a vector \mathbf{x} and a matrix $\mathbf{A} \equiv [\mathbf{a}_1, \mathbf{a}_2, \dots]$, where \mathbf{a}_k ($k = 1, 2, \dots$) are the column vectors of \mathbf{A} , then it can be shown that $\text{diag}(\mathbf{x}) \mathbf{A} = [\mathbf{x}, \mathbf{x}, \dots] \dot{\times} \mathbf{A} = [\mathbf{x} \dot{\times} \mathbf{a}_1, \mathbf{x} \dot{\times} \mathbf{a}_2, \dots]$, where $\dot{\times}$ stands for element-wise (Hadamard) product.

2.3 Localization in the C-GIES algorithm

As aforementioned, one of the core ideas used in this work to derive constrained assimilation algorithms is the notion of “perfect measurements” [34], where the constraint systems are treated as forward observation operators without any measurement errors. In the context of ensemble data assimilation, localization is an auxiliary technique often adopted to mitigate the problem of ensemble collapse and strengthen the performance of assimilation algorithms [7, 10, 28]. When applying localization to “perfect measurements” from the constraint systems, however, there are two issues worthy of special attention.

One issue is that the “perfect measurements” might not possess any physical locations, such that (physical) distance-based localization methods [7, 10] may not be applicable. In fact, as one may notice in the C-GIES update formula (32), or its special case (33), in general it is the gradients $\nabla_{\mathcal{D}_d} [\mathcal{T}_d(\mathbf{d}_j) - \mathcal{T}_d \circ \mathbf{g}(\mathbf{m}_j^i)]$, $\nabla_{\mathcal{D}_{eq}} [\mathcal{T}_{eq}(\mathbf{0}) - \mathcal{T}_{eq} \circ \mathbf{f}_{eq}(\mathbf{m}_j^i)]$ and $\nabla_{\mathcal{D}_{in}} [\mathcal{T}_{in}(\mathbf{0}) - \mathcal{T}_{in} \circ \mathbf{h}_{in}(\mathbf{m}_j^i)]$ that contribute to model updates, similar to the role of normalized data residual $\mathbf{C}_d^{-1} [\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^i)]$ in the original IES algorithm (cf. Eqs. 4 and 5).³ As such, for the umbrella GIES algorithm or its descendant C-GIES algorithm for DASC problems, in general it would be more appropriate to exploit the gradients of respective observation or constraint systems, rather than the locations of observations, for localization.

Another problem is that even though the number of real observations (e.g., production data) may not be big, the dimensions of the constraint systems could be very high. As an example, one may consider box constraints imposed on all petro-physical parameters distributed on reservoir gridblocks, which is the experiment setting to be adopted in one of the case studies later. In large scale history matching problems, a straightforward application of the conventional localization technique would thus consume a huge amount of computer memory, since it involves a matrix (called tapering matrix hereafter) in the dimension of model size by effective data size, where by “effective data size” we mean the number of real measurements plus the dimensions of constraint systems.

³In fact, it can be shown that $\mathbf{C}_d^{-1} [\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^i)]$ is the gradient of the particular choice of data mismatch term $\mathcal{D}(\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1})) = \frac{1}{2} (\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}))^T \mathbf{C}_d^{-1} (\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^{i+1}))$ in the original IES algorithm.

To simultaneously tackle the aforementioned two problems, namely, (1) localization based on information of gradients, rather than physical locations of real and “perfect” measurements; and (2) localization involving big models and big effective data induced by constraints, there are two possibly strategies. One strategy is to follow the recent work of [37], which extends correlation-based Kalman-gain type localization [24, 28] to correlation-based local analysis. In the context of C-GIES, one can compute the sample correlations between an ensemble of reservoir models and the corresponding ensemble of terms related to the gradients $(\nabla_{\mathcal{D}_d} [\mathcal{T}_d(\mathbf{d}_j) - \mathcal{T}_d \circ \mathbf{g}(\mathbf{m}_j^i)], \nabla_{\mathcal{D}_{eq}} [\mathcal{T}_{eq}(\mathbf{0}) - \mathcal{T}_{eq} \circ \mathbf{f}_{eq}(\mathbf{m}_j^i)]$ and/or $\nabla_{\mathcal{D}_{in}} [\mathcal{T}_{in}(\mathbf{0}) - \mathcal{T}_{in} \circ \mathbf{h}_{in}(\mathbf{m}_j^i)]$). Using the information of these correlations would eliminate the need for physical locations and render a few additional practical advantages, e.g., the ability to handle both non-local observations and the spatial-temporal characteristics of observations, see the elaborations in [24, 28]. Meanwhile, local analysis can be employed to tackle the issue of big model and big datasets. Instead of performing a single global update, local analysis conducts a series of local updates, each of them only involving a small number of model variables and effective data that are grouped based on the computed sample correlations [37].

Another practical strategy is still based on correlation-based Kalman-gain localization. To deal with the presence of big datasets in localization, however, we do not directly handle the big datasets themselves. Instead, we follow the idea in [29] and project the big datasets onto certain ensemble subspaces. After projections, we apply correlation-based localization to terms related to the projected gradients, e.g., $(\mathbf{S}_g^i)^T \mathbf{C}_d^{-1} [\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^i)], (\mathbf{S}_{f_{eq}}^i)^T \nabla_{\mathcal{D}_{eq}} [\mathbf{0} - \mathbf{f}_{eq}(\mathbf{m}_j^i)],$ and $(\mathbf{S}_{h_{in}}^i)^T \nabla_{\mathcal{D}_{in}} [\mathbf{0} - \mathbf{h}_{in}(\mathbf{m}_j^i)]$ in Eq. 33. All these projected gradients are column vectors in the dimension of $N_e \times 1$, with N_e being the ensemble size, hence the sizes of projected gradients are independent of the dimensions of the original gradients. One more issue to notice here is that in Eq. 33, the term

$$\mathbf{F}_j^i \equiv \left[(\mathbf{S}_g^i)^T \mathbf{C}_d^{-1} \mathbf{S}_g^i + \alpha_j^i (\mathbf{S}_{f_{eq}}^i)^T \nabla_{\mathcal{D}_{eq}}^2 [\mathbf{0} - \mathbf{f}_{eq}(\mathbf{m}_j^i)] \right. \\ \left. \mathbf{S}_{f_{eq}}^i + \beta_j^i (\mathbf{S}_{h_{in}}^i)^T \nabla_{\mathcal{D}_{in}}^2 [\mathbf{0} - \mathbf{h}_{in}(\mathbf{m}_j^i)] \mathbf{S}_{h_{in}}^i + \gamma_j^i \mathbf{I}_{N_e} \right] d^{-1} \tag{42}$$

in general depends on individual ensemble members \mathbf{m}_j^i . To separate the effects of observed data and constraints on model updates, at each iteration step i , we choose to multiply all the projected gradients by this matrix factor \mathbf{F}_j^i (in the dimension of $N_e \times N_e$), and then use the ensemble

of the following products (all in the dimension of $N_e \times 1$)

$$\mathbf{A}_{d,j}^i \equiv \mathbf{F}_j^i (\mathbf{S}_g^i)^T \mathbf{C}_d^{-1} [\mathbf{d}_j - \mathbf{g}(\mathbf{m}_j^i)]; \tag{43}$$

$$\mathbf{A}_{eq,j}^i \equiv \mathbf{F}_j^i (\mathbf{S}_{f_{eq}}^i)^T \nabla_{\mathcal{D}_{eq}} [\mathbf{0} - \mathbf{f}_{eq}(\mathbf{m}_j^i)]; \tag{44}$$

$$\mathbf{A}_{in,j}^i \equiv \mathbf{F}_j^i (\mathbf{S}_{h_{in}}^i)^T \nabla_{\mathcal{D}_{in}} [\mathbf{0} - \mathbf{h}_{in}(\mathbf{m}_j^i)], \tag{45}$$

together with the ensemble of reservoir models \mathbf{m}_j^i (for $j = 1, 2, \dots, N_e$) to compute three sets of correlation fields, denoted by $\mathbf{C}_d^i, \mathbf{C}_{eq}^i$ and \mathbf{C}_{in}^i (all in the dimension of $m \times N_e$), respectively. Then following the previous work [24, 28], we can construct three tapering matrices, denoted by $\mathbf{T}_d^i, \mathbf{T}_{eq}^i$ and \mathbf{T}_{in}^i (all in the dimension of $m \times N_e$), based on $\mathbf{C}_d^i, \mathbf{C}_{eq}^i$ and \mathbf{C}_{in}^i . These tapering matrices are applied to the model update formula (33), such that the model update formula with correlation-based localization reads as follows:

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + (\mathbf{T}_d^i \dot{\times} \mathbf{S}_L^i) \mathbf{A}_{d,j}^i + \alpha_j^i (\mathbf{T}_{eq}^i \dot{\times} \mathbf{S}_L^i) \mathbf{A}_{eq,j}^i \\ + \beta_j^i (\mathbf{T}_{in}^i \dot{\times} \mathbf{S}_L^i) \mathbf{A}_{in,j}^i. \tag{46}$$

In the current work, we adopt the projection-based strategy to implement correlation-based Kalman-gain localization, as this requires minimal changes of the C-GIES update formula in Eqs. 32 or 33, in contrast to the strategy based on correlation-based local analysis. Throughout this work, we use the localization scheme, labeled as RndShfl-GC, in [24] to construct the tapering matrices $\mathbf{T}_d^i, \mathbf{T}_{eq}^i$ and \mathbf{T}_{in}^i . For brevity, however, we skip the details of RndShfl-GC, and refer readers to [24] for more information.

3 Numerical results

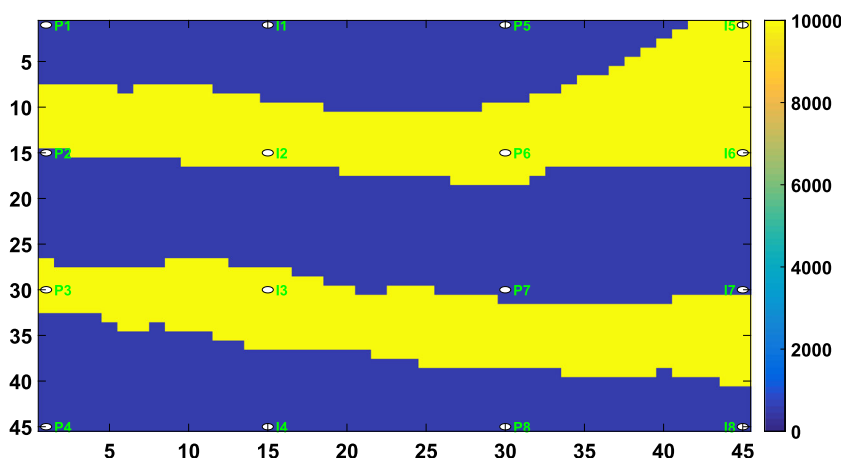
3.1 DASC problem in a 2D case study

3.1.1 Experiment settings of the 2D case study

This section focuses on illustrating the performance of the chosen C-GIES algorithm, (33), in a 2D channelized reservoir characterization problem, which was investigated in the previous work [23]. The size of the 2D reservoir model is 45×45 , in terms of the number of gridblocks. The unknown parameters of the reservoir model contain x -dimensional permeability (PERMX) distributed on all reservoir gridblocks, whereas other parameters (e.g., porosity) are assumed to be known. As such, the total number of parameters to be estimated in this 2D case study is 2025.

Figure 1 indicates the reference PERMX map, where the positions of 8 injectors (I1 – I8) and 8 producers (P1 – P8) are marked. The reference model contains two facies, sand and shale, with their corresponding PERMX

Fig. 1 Reference model used in the 2D synthetic case study, with well positions marked by white ellipses. Well names (in green) starting with the letter “I” stand for injection wells, while those with the letter “P” for production wells



values being 10000md and 500md , respectively. The reference reservoir model is simulated using ECLIPSE[®] 100 black oil simulator, with a total simulation time of 3800 days. The production data generated during the first 1900 days (history-matching period), including well oil/water production rates (WOPR/WWPR) from 8 producers, and well bottom-hole pressures (WBHP) from 8 injectors, are used to history-match an ensemble of reservoir models, whereas the corresponding production data during the second 1900 days (forecast period) are reserved to cross-validate the performance of different history matching algorithms. The total numbers of production data during history matching and forecast periods are both 240. To mimic the situation in practical history matching problems, we add (zero-mean) Gaussian white noise to all the production data during the history matching period. The standard deviations (STDs) of the noise in both WOPR and WWPR data are $0.2236\text{ m}^3/\text{day}$, while those of the noise in WBHP data are 0.2646 bar . On the other hand, during the forecast period, no noise is added to the production data.

An ensemble of 100 PERMX maps is generated by the SNESIM algorithm [38] to initialize ensemble-based history matching algorithms. Figure 2 shows the mean and STD maps of the initial ensemble.

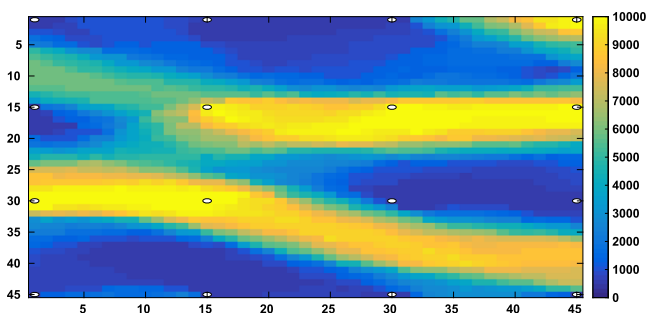
In this case study, we consider one IN- and one EQ-constraint systems. The IN-constraint system imposes a box constraint on the range of the PERMX value m_k on the k th gridblock ($k = 1, 2, \dots, 2025$), such that $100\text{md} \leq m_k \leq 15000\text{md}$. This box constraint then leads to the following IN-constraint system

$$100 - m_k \leq 0; \tag{47}$$

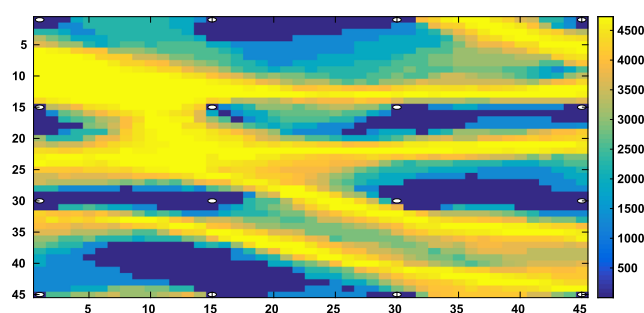
$$m_k - 15000 \leq 0, \tag{48}$$

for $k = 1, 2, \dots, 2025$. As a result, the dimension of the IN-constraint system is $2 \times 2025 = 4050$.

The EQ-constraint system aims to impose a constraint on the shape of the histogram of an estimated PERMX map. To this end, we assume that the ratio of sand and shale facies in the reference PERMX map (cf Fig. 1), hence the corresponding histogram, is (roughly) known (e.g., based on well log data). The first row of Fig. 6 shows the histogram of the reference PERMX map and that corresponding to the mean of the initial ensemble of PERMX maps. During history matching, we compare the histogram of each estimated PERMX map to that of the reference PERMX map bin by bin. For a quantitative comparison, in each bin, we count the number of permeability values falling into that bin, while the total number of bins is 50. For convenience of



(a) Mean of the initial ensemble



(b) STD of the initial ensemble

Fig. 2 Mean (left) and STD (right) maps of the initial ensemble of PERMX maps

discussion later, we denote by \mathbf{m}_{ref} the reference PERMX map, and by \mathcal{S}_H the operator that transforms a generic PERMX map \mathbf{m} ($\mathbf{m} = [m_1, m_2, \dots, m_{2025}]^T$) to a set of numbers, through the steps: PERMX map \rightarrow histogram \rightarrow number of permeability values in each bin. Following this setting, the EQ-constraint system is composed as follows:

$$\mathcal{S}_H(\mathbf{m}) - \mathcal{S}_H(\mathbf{m}_{ref}) = \mathbf{0}, \tag{49}$$

whereas the dimension of the EQ-constraint system equals the number of bins of a histogram (50 in this case). Here, we note that due to the possible conflict with other objectives (e.g., that of reducing data mismatch), these EQ-constraints may not be strictly satisfied during the history matching process. Nevertheless, as will be shown later, the presence of these EQ-constraints does tend to help improve the performance of history matching.

Here we compare the performance of a few IES algorithms. The base case for comparison is the original IES algorithm (cf Eqs. 4–8) combined with a simple truncation strategy to handle a box-constraint $m_{lb} \leq m_k \leq m_{ub}$ ($k = 1, 2, \dots, 2025$ in the current case study), where m_{lb} ($= 100$) and m_{ub} ($= 15000$) represent lower and upper bounds, respectively, for a model variable m_k . With the truncation strategy, whenever an m_k estimated by the original IES algorithm is less than m_{lb} , then we set $m_k = m_{lb}$; Likewise, whenever an estimate m_k is larger than m_{ub} , then we set $m_k = m_{ub}$. For reference later, the base-case algorithm is referred to as O-IES.

The other algorithms in comparison include the C-GIES algorithms (cf. Eq. 33) that take into account the IN-constraints (47–48) and/or the EQ-constraints (49). For distinction later, we refer to these algorithms as C-GIES-IN, C-GIES-EQ and C-GIES-(IN+EQ), respectively. By “C-GIES-IN”, we mean that the C-GIES algorithm only deals with the IN-constraint system. Similarly, the C-GIES-EQ algorithm only considers the EQ-constraint system, and the C-GIES-(IN+EQ) algorithm takes into account both constraint systems. Note that all these three types of C-GIES algorithms are also equipped with the truncation strategy, such that the differences of history-matching performance among these IES algorithms would reflect the impacts of the constraints incorporated through our proposed framework.

The C-GIES algorithms based on Eq. 33 require to determine the weight coefficients α_j^i , β_j^i and γ_j^i at each iteration step. In the current work, we follow the strategy in [25], and set α_j^i as follows:

$$\alpha_j^i = w_1 \text{trace} \left(\left(\mathbf{S}_g^i \right)^T \mathbf{C}_d^{-1} \mathbf{S}_g^i \right) / \text{trace} \left(\left(\mathbf{S}_{f_{eq}}^i \right)^T \nabla_{\mathcal{D}_{eq}}^2 \left[\mathbf{0} - \mathbf{f}_{eq} \left(\mathbf{m}_j^i \right) \right] \mathbf{S}_{f_{eq}}^i \right), \text{ if with EQ-constraints; } \tag{50}$$

$$\alpha_j^i = 0, \text{ if without EQ-constraints, } \tag{51}$$

where w_1 is a positive constant not too far away from 1, and the trace operator calculates the trace of a matrix. The rationale behind Eq. 50 is that when the EQ-constraint system is present, we want its influence on model update, in terms of trace $\left(\left(\mathbf{S}_{f_{eq}}^i \right)^T \nabla_{\mathcal{D}_{eq}}^2 \left[\mathbf{0} - \mathbf{f}_{eq} \left(\mathbf{m}_j^i \right) \right] \mathbf{S}_{f_{eq}}^i \right)$, to be comparable to that of the data mismatch, in terms of trace $\left(\left(\mathbf{S}_g^i \right)^T \mathbf{C}_d^{-1} \mathbf{S}_g^i \right)$. The value of β_j^i is determined in a similar way, as follows:

$$\beta_j^i = w_2 \text{trace} \left(\left(\mathbf{S}_g^i \right)^T \mathbf{C}_d^{-1} \mathbf{S}_g^i \right) / \text{trace} \left(\left(\mathbf{S}_{h_{in}}^i \right)^T \nabla_{\mathcal{D}_{in}}^2 \left[\mathbf{0} - \mathbf{h}_{in} \left(\mathbf{m}_j^i \right) \right] \mathbf{S}_{h_{in}}^i \right), \text{ if with IN-constraints; } \tag{52}$$

$$\beta_j^i = 0, \text{ if without IN-constraints, } \tag{53}$$

where w_2 is another positive constant not too far away from 1. The values of (w_1, w_2) adopted for the four algorithms in comparison, namely, O-IES, C-GIES-IN, C-GIES-EQ and C-GIES-(IN+EQ), can be found in Table 1.

As for the regularization parameter γ_j^i , it is calculated as follows:

$$\gamma_j^i = w_3^i \text{trace} \left(\left(\mathbf{S}_g^i \right)^T \mathbf{C}_d^{-1} \mathbf{S}_g^i + \alpha_j^i \left(\mathbf{S}_{f_{eq}}^i \right)^T \nabla_{\mathcal{D}_{eq}}^2 \times \left[\mathbf{0} - \mathbf{f}_{eq} \left(\mathbf{m}_j^i \right) \right] \mathbf{S}_{f_{eq}}^i + \beta_j^i \left(\mathbf{S}_{h_{in}}^i \right)^T \nabla_{\mathcal{D}_{in}}^2 \times \left[\mathbf{0} - \mathbf{h}_{in} \left(\mathbf{m}_j^i \right) \right] \mathbf{S}_{h_{in}}^i \right) / N_e. \tag{54}$$

Note that in Eq. 54 we have used the fact that $\text{trace}(\mathbf{I}_{N_e}) = N_e$, where w_3^i is a positive scalar that adaptively changes over the iteration step. Following [25], we let $w_3^0 = 1$, $w_3^{i+1} = 0.9 \times w_3^i$ if the average data mismatch over ensemble members (to be defined later) at the $(i + 1)$ th iteration step is lower than that at the i th iteration step; otherwise, we set $w_3^{i+1} = 2 \times w_3^i$. During history matching, the maximum number of the iteration steps for all algorithms is 50, but the iteration process may stop earlier, if the change of the average data mismatch values during two consecutive iteration steps is less than 0.01%.

The use of the distance metrics \mathcal{D}_{in} and \mathcal{D}_{eq} also requires to specify constants \mathbf{a} , \mathbf{b} and ϵ , cf. Eqs. 34–41. In the experiments, we set $\mathbf{a} = \mathbf{b} = 0.1 \times \mathbf{1}$ and $\epsilon = 0.001$.

To cross-validate the performance of history matching algorithms, we adopt the following two measures, namely, forecast data mismatch and root mean square error (RMSE). Data mismatch uses a weighted ℓ_2^2 metric in the data space to measure the distance between the observed and the simulated data. Given an ensemble $\mathcal{M}^i = \{\mathbf{m}_j^i\}_{j=1}^{N_e}$ of reservoir models, a forward reservoir simulator \mathbf{g} and

Table 1 Performance of the four history-matching algorithms in the 2D case study. The performance is measured in terms of data mismatch (mean ± STD) during the forecast period, and RMSE (mean ± STD), which are computed using the ensembles of reservoir models at the final iteration steps

	History-matching data mismatch (mean ± STD) × 10 ³	Forecast data mismatch (mean ± STD) × 10 ³	RMSE of PERMX (mean ± STD) × 10 ³	Values of (w ₁ , w ₂)
Initial ensemble	5.7936 ± 2.2513	7.5940 ± 3.7789	5.2665 ± 0.4404	(N/A, N/A)
O-IES	2.2367 ± 0.8683 (−61.3936%)	5.2670 ± 2.5087 (−30.6426%)	4.5398 ± 0.3245 (−13.7985%)	(0, 0)
C-GIES-EQ	2.3015 ± 0.9678 (−60.2751%)	4.5766 ± 2.2940 (−39.7340%)	4.4773 ± 0.3036 (−14.9853%)	(1, 0)
C-GIES-IN	2.5579 ± 1.0900 (−55.8496%)	5.1431 ± 2.1756 (−32.2742%)	4.4380 ± 0.2916 (−15.7315%)	(0, 1)
C-GIES-(IN+EQ)	2.1575 ± 0.6781 (−62.7606%)	4.0826 ± 1.2078 (−46.2391%)	3.9053 ± 0.1863 (−25.8464%)	(0.5, 0.5)

For comparison, the quantities with respect to the initial ensemble are also included, and the relative changes of the average quantities between the initial and final ensembles are provided in separate parentheses. In addition, we note that the values of (w₁, w₂) are not applicable (N/A) to the initial ensemble

the observed field dataset **d**^o, an ensemble $\mathcal{E}(\mathcal{M}^i)$ of data mismatch values is computed as follows:

$$\mathcal{E}(\mathcal{M}^i) \equiv \left\{ \Phi_j^i | \Phi_j^i = \left(\mathbf{d}^o - \mathbf{g}(\mathbf{m}_j^i) \right)^T \mathbf{C}_d^{-1} \left(\mathbf{d}^o - \mathbf{g}(\mathbf{m}_j^i) \right) \right\}_{j=1}^{N_e}, \tag{55}$$

where **C**_d is the observation error covariance matrix associated with the dataset **d**^o (also cf. Eq. 3 and the corresponding text for the definition of **C**_d).

Similarly, RMSE adopts a (scaled) ℓ_2 metric in the model space to examine the distance between the reference and the estimated reservoir models. Given an *m*-dimensional reference model **m**^{ref} and an ensemble $\mathcal{M}^i = \{\mathbf{m}_j^i\}_{j=1}^{N_e}$ of reservoir models, an ensemble $\Omega(\mathcal{M}^i)$ of RMSE is calculated as follows:

$$\Omega(\mathcal{M}^i) \equiv \left\{ \zeta_j^i | \zeta_j^i = \frac{\|\mathbf{m}_j^i - \mathbf{m}^{ref}\|_2}{\sqrt{m}} \right\}_{j=1}^{N_e}. \tag{56}$$

3.1.2 Numerical results of the 2D case study

Table 1 reports the data mismatch values (mean ± STD) during both history-matching and forecast periods, and RMSE values (mean ± STD), with respect to both the initial ensemble, and the final ensembles obtained by four history-matching algorithms: O-IES, C-GIES-IN, C-GIES-EQ and C-GIES-(IN+EQ). The values of the pair (w₁, w₂) used by these history-matching algorithms are also listed therein. In comparison to the initial ensemble, all four algorithms are able to achieve better history-matching performance, in terms of both lower forecast data mismatch and RMSE values. In addition, compared with the base-case O-IES algorithm, the three C-GIES algorithms achieve enhanced performance, by exploiting either EQ-constraints or IN-constraints, or both. As a matter of fact, in this particular case study, it is the C-GIES-(IN+EQ) algorithm with both types of constraints that achieves the best performance in terms of both forecast data mismatch and RMSE, manifesting the benefits of incorporating both types of constraints into history matching through the proposed workflow.

Table 2 Values of barrier and channel functions (cf. Eqs. 34 and 37) in the 2D case study, evaluated with respect to the initial ensemble, and the final ensembles obtained by the O-IES, C-GIES-IN, C-GIES-EQ and C-GIES-(IN+EQ) algorithms

	Value of barrier function (for IN-constraint system) (mean ± STD) × 10 ⁴	Value of channel function (for EQ-constraint system) (mean ± STD)
Initial ensemble	- 3.3269 ± 0.0217	- 100.6942 ± 27.6959
O-IES	- 3.1669 ± 0.0903 (+4.8093%)	149.2952 ± 14.7206 (+ 248.2659 %)
C-GIES-EQ	- 3.1696 ± 0.0662 (+4.7281%)	144.5890 ± 17.1123 (+ 243.5922 %)
C-GIES-IN	- 3.2240 ± 0.0895 (+3.0930%)	150.9701 ± 13.4692 (+ 249.9293 %)
C-GIES-(IN+EQ)	- 3.2427 ± 0.0522 (+2.5309%)	157.7672 ± 6.8273 (+ 256.6795%)

As in Table 1, the relative changes of the average values of barrier and channel functions between the initial and final ensembles are provided in separate parentheses

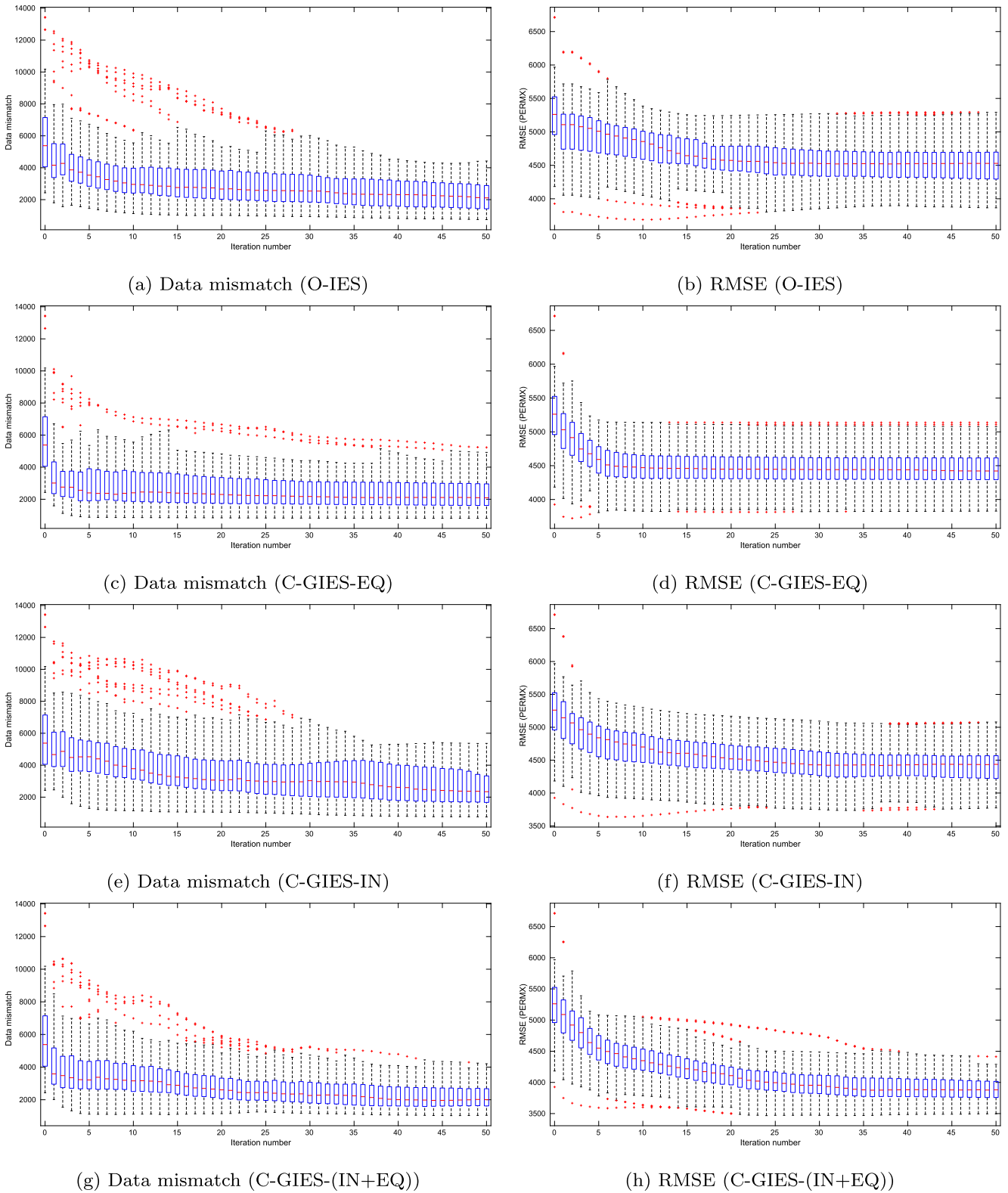


Fig. 3 Box plots of data mismatch during the history-matching period (left column) and corresponding RMSE values (right column) at different iteration steps of the O-IES, C-GIES-IN, C-GIES-EQ and C-GIES-IN+EQ algorithms

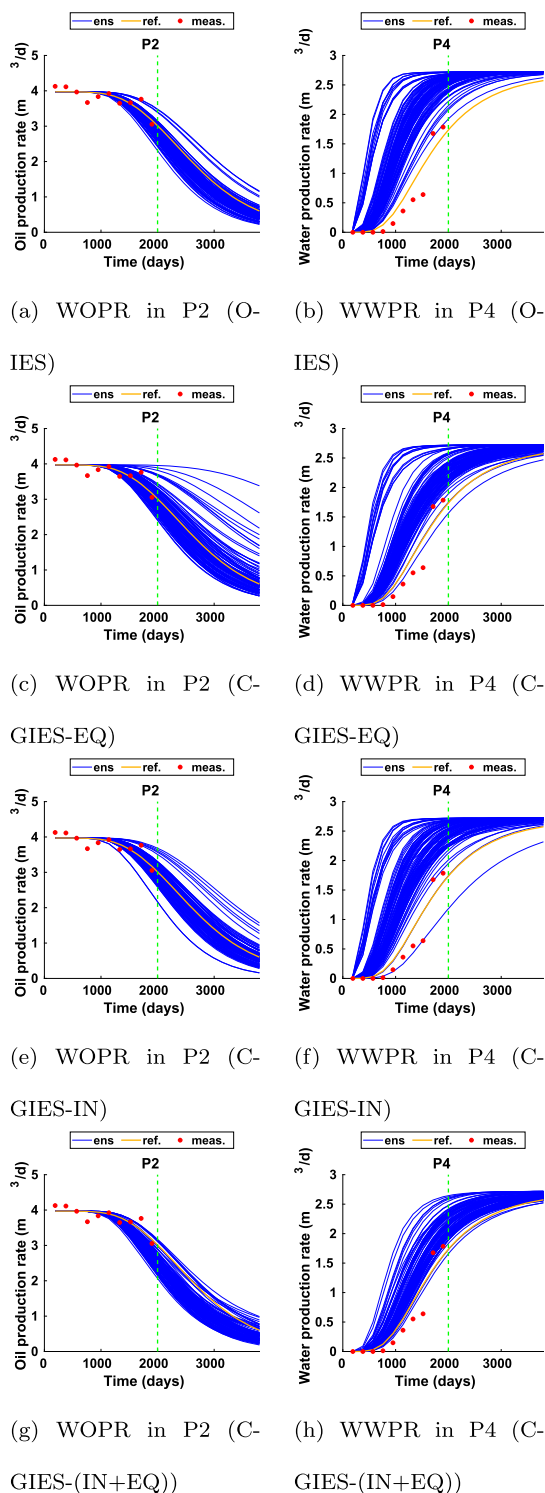


Fig. 4 Profiles of WOPR in P2 (left column) and WWPR in P4 (right column) in the 2D case study. In each sub-figure, the orange curve stands for the production data generated by the reference reservoir model without observation errors; the red dots for noisy observations used for history matching; the blue curves for the simulated production data using the ensembles obtained at the last iteration steps of the O-IES, C-GIES-IN, C-GIES-EQ and C-GIES-(IN+EQ) algorithms; and the green vertical dashed line separates the history-matching and forecast periods

Similarly, Table 2 shows the values (mean \pm STD) of barrier and channel functions (cf. Eqs. 34 and 37), with respect to the initial ensemble, and the final ensembles obtained by the four history-matching algorithms. Note that these function values are calculated only at the post-history-matching stage, and are not utilized during history matching. Compared to the initial ensemble, one can see that the final ensembles obtained by the four history-matching algorithms end up with higher barrier and channel function values, which is different from the situation in Table 1, wherein the final ensembles achieve lower data mismatch values. This contrast suggests that reducing data mismatching and enforcing EQ- and/or In-constraints could be conflicting objectives. On the other hand, by comparing the base-case algorithm with the three C-GIES algorithms, the benefits of incorporating EQ- and/or In-constraints through the proposed workflow will be better appreciated. For instance, after incorporating the EQ-constraints, one can see that the channel function values of the C-GIES-EQ algorithm tend to be noticeably lower than those of the O-IES algorithm; Likewise, after incorporating the IN-constraints, the barrier function values of the C-GIES-IN algorithm also tend to be lower than those of the O-IES algorithm. A more complicated situation happens in the C-GIES-(IN+EQ) algorithm, wherein the barrier function values, as expected, tend to be lower than those of the O-IES algorithm, but the channel function values, surprisingly, tend to be higher than those of the O-IES algorithm. A possible explanation to this situation is that in this particular case study, the objective of honoring the box-constraints may also conflict with the objective of changing the shape of a histogram.

For further illustration, Fig. 3 presents the box plots of data mismatch values (left column) computed using the ensembles of reservoir models at different iteration steps of the O-IES, C-GIES-IN, C-GIES-EQ and C-GIES-(IN+EQ) algorithms; and the corresponding RMSE values (right column) at different iteration steps. As one can see from these plots, both data mismatch and RMSE values tend to decrease as the iteration process goes on. In consistency with the results in Table 1, it is the last ensemble obtained by the C-GIES-(IN+EQ) algorithm that tends to achieve the lowest RMSE values.

Figure 4 shows the production-data profiles, with respect to WOPR in P2 and WWPR in P4. The forecast production data (blue curves) are computed using the final ensembles of the four history-matching algorithms. Overall, for WOPR in P2, the forecast production data generated by the four history-matching algorithms match the reference (orange curve) reasonably well. For WWPR in P4, however, higher data mismatch is spotted. Among the four history-matching algorithms, the C-GIES-(IN+EQ) algorithm has the best performance, as is reported in Table 1.

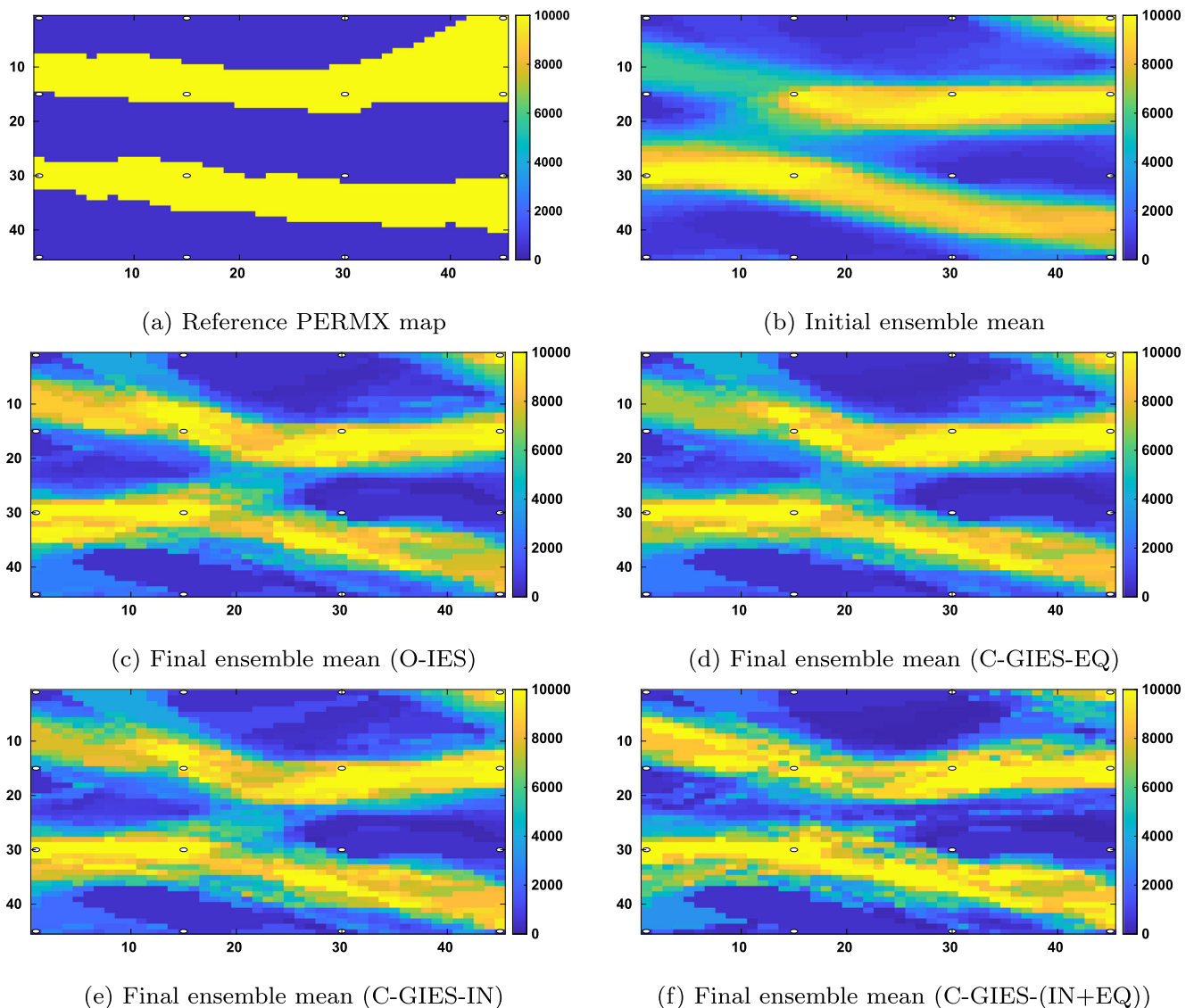


Fig. 5 Mean PERMX mps of the final ensembles obtained by the four history-matching algorithms in the 2D case study. For comparison, the reference PERMX map and the mean map of the initial ensemble are re-plotted here

In Fig. 5, we report the mean PERMX maps of the final ensembles obtained by the four history-matching algorithms, as well as the reference PERMX map and the mean map of the initial ensemble. All these four algorithms do not perform well in term of preserving the connectivities of the channel bodies, although all of them appear to be able to capture the main geological structures in the reference map. The history-matching performance is expected to be further improved by adopting a certain more sophisticated model re-parameterization strategy, e.g., through deep unsupervised learning [5]; or by incorporating into the cost function (e.g., in Eq. 11) a more suitable regularization term [23]. Either investigation, however, is beyond the scope of the current work.

Finally, Fig. 6 compares a set of histograms calculated from the reference PERMX map, the mean maps of the initial ensemble and the final ensembles obtained by the four history-matching algorithms. In line with the results in Table 2, the histogram of the C-GIES-(IN+EQ) algorithm appears to be more different from the others, e.g., apart from the two modes located at the correct values 500 and 10000 (as in the reference histogram), that histogram has two additional peaks with substantial heights whose locations are inconsistent with those in the reference histogram. In contrast, the histograms of the O-IES, C-GIES-IN, and C-GIES-EQ algorithms roughly have bi-modal distributions, and exhibit more similarities to those of the reference PERMX map.

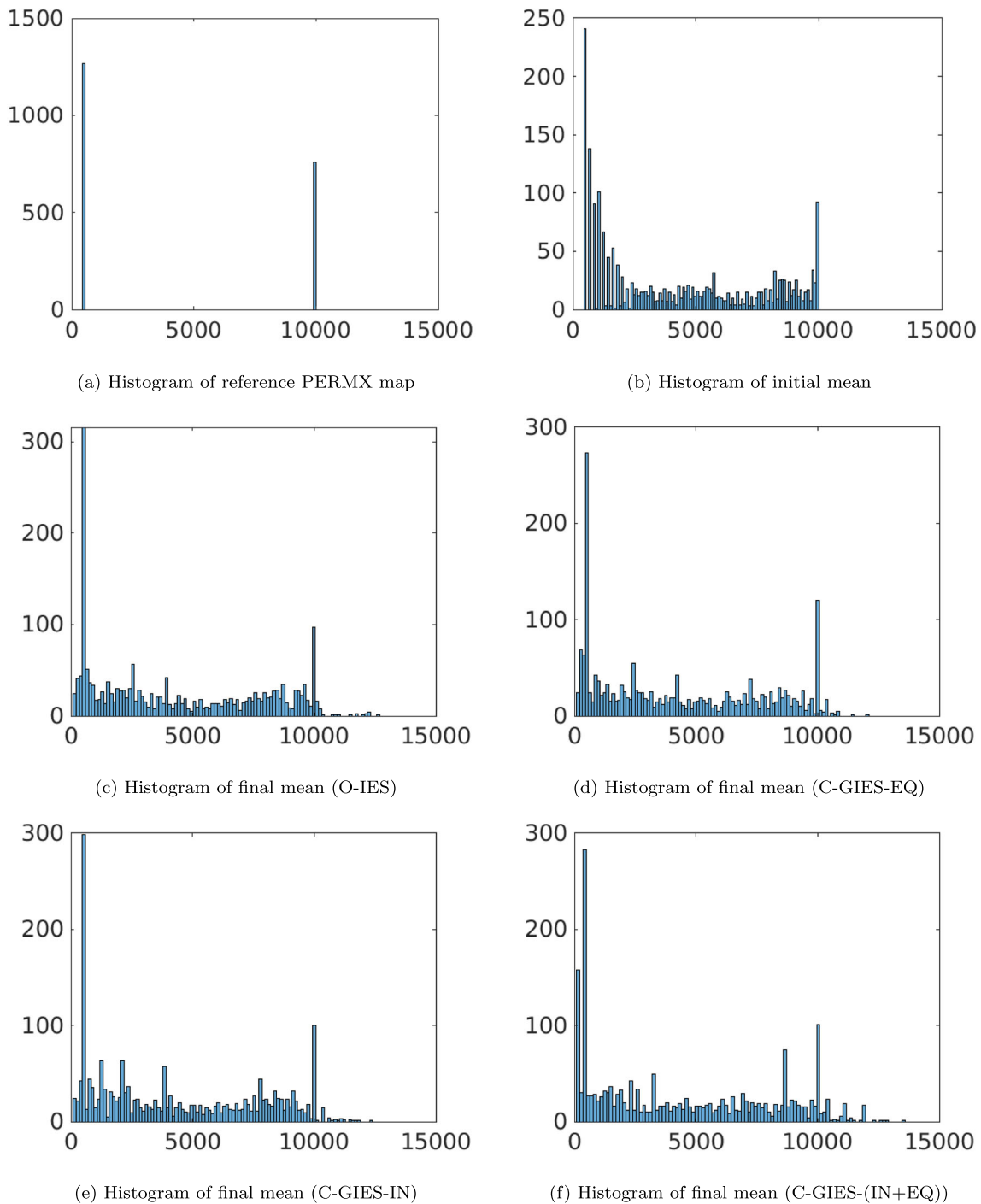


Fig. 6 Histograms of the mean PERMX maps with respect to the initial ensemble and the final ones obtained by the four history-matching algorithms. The histogram of the reference PERMX map is also included for comparison

3.2 DASC problem in a 3D case study

3.2.1 Experiment settings of the 3D case study

In the second numerical example, we apply IES algorithms to the Brugge benchmark case [31], in which the reservoir model is 3D, in the dimension of $139 \times 48 \times 9$. Therefore,

there are 60048 gridblocks in total, and 44550 of these gridblocks are active. The parameters to be estimated include x -, y - and z -dimensional permeability values (denoted by PERMX, PERMY, PERMZ, respectively) in the scale of natural logarithm, and porosity values (denoted by PORO) distributed on these 44550 active gridblocks. Hence, the total number of unknown parameters is $4 \times 44550 =$

178200. In this particular case study, the histograms of the parameters do not appear to provide much information of the geological structures, and it does not seem to be realistic to assume that we know the histograms of the reference reservoir model. As such, here we do not consider histogram-based equality constraints as in the previous 2D example. Instead, we only consider a set of box constraints imposed on PERMX, PERMY, PERMZ and PORO values. Specifically, for PERMX values, the box constraints are that $-4 \leq \ln(m_{permx}) \leq 9.2$, where \ln is the natural logarithmic function, m_{permx} represents the PERMX value on an active gridblock; Similarly, the box constraints for other types of petro-parameters are $-4 \leq \ln(m_{permy}) \leq 9.2$; $-7 \leq \ln(m_{permz}) \leq 6.9$ and $0.07 \leq m_{poro} \leq 0.3$. As a result, the IN-constraint system is almost the same as that in the 2D case system, except that the lower and upper bounds, and the dimensionality are different. In the current case study, the dimension of the IN-constraint system is $2 \times 178200 = 356400$, twice the reservoir-model size. To handle the high dimensionality of the IN-constraint system, the diagonal form of the Hessian matrix in Eq. 36 proves to be particularly useful for reducing the consumption of computer memory, otherwise it becomes prohibitively expensive to directly manipulate the Hessian matrix in the dimension of 356400×356400 (which requires around 900 GB memory in MATLAB®).

The initial ensemble of reservoir models is taken from the Brugge benchmark dataset, which contains 104 ensemble members. One of the members is chosen as the reference model to generate observed production data at 20 producers (labeled as BR-P-1, BR-P-2, ..., BR-P-20, or simply as P1, P2, ..., P20) and 10 injectors (labeled as BR-I-1, BR-I-2, ..., BR-I-10, or simply as I1, I2, ..., I10). The production period

is 10 years, and for history matching we use the following types of production data reported at 20 time steps: well oil production rates (WOPR) and water cuts (WWCT) from the 20 producers, and the bottom hole pressures (WBHP) from all 30 wells. As a result, the total number of production data is 1400. The production data are contaminated by a certain amount of zero-mean Gaussian white noise. For WOPR and WWCT data, the STDs of the noise components are 10% of the magnitudes of the pure production data generated by the reference model. In case that the magnitude of a data point is zero, then the STD is set to be 10^{-6} instead, to avoid the possible issue of division by zero when calculating data mismatch values. On the other hand, for WBHP data, the STDs of the noise components are all set to 1 bar. In this case study, we use all available data for history matching, so there is no forecast period. As such, we adopt RMSE for performance cross-validation.

In the sequel, we compare the performance of two history-matching algorithms. One algorithm is the original IES, but combined with the truncation strategy to honor the box constraints. This algorithm is again referred to as O-IES, as in the 2D case study. Another algorithm is a C-GIES with the IN-constraint system induced by the box constraints. Following the custom in the 2D case study, this algorithm is referred to as C-GIES-IN, and we note that here the C-GIES-IN is also equipped with the truncation strategy.

Since there is no EQ-constraint system involved, the weight coefficient w_1 (cf. Eq. 50) is set to 0; the weight coefficient w_2 (cf. Eq. 52) associated with the IN-constraint system is set to 5; and the weight coefficient w_3^i (cf. Eq. 54) will adapt to the iteration steps, following the same rule as in the 2D case study. During history matching, the maximum number of the iteration steps for O-IES and C-GIES-IN

Table 3 Performance of the two history-matching algorithms in the Brugge case study. The performance is measured in terms of RMSE (mean \pm STD), which are computed using the ensembles of reservoir models at the first and final iteration steps

	Initial ensemble	O-IES	C-GIES-IN
Data mismatch	$3.6232 \times 10^9 \pm 1.4900 \times 10^{10}$	$(3.9616 \pm 2.9947) \times 10^7$ (−98.9066%)	$(7.0091 \pm 5.5507) \times 10^6$ (−99.8065%)
Value of barrier function	$-3.4172 \times 10^5 \pm 6.6936 \times 10^3$	$-3.4217 \times 10^5 \pm 5.9683 \times 10^3$ (−0.1317%)	$-3.4258 \times 10^5 \pm 3.9202 \times 10^3$ (−0.2517%)
RMSE (PERMX)	1.6585 ± 0.3827	1.4167 ± 0.2545 (−14.5794%)	1.4119 ± 0.2284 (−14.8689%)
RMSE (PERMY)	1.6612 ± 0.3794	1.4198 ± 0.2515 (−14.5317%)	1.4133 ± 0.2244 (−14.9229%)
RMSE (PERMZ)	2.0077 ± 0.4096	1.8054 ± 0.3101 (−10.0762%)	1.7636 ± 0.2916 (−12.1582%)
RMSE (PORO)	0.0302 ± 0.0025	0.0280 ± 0.0025 (−7.2848%)	0.0285 ± 0.0028 (−5.6291%)
RMSE (all together)	1.5450 ± 0.3362	1.3498 ± 0.2344 (−12.6343%)	1.3327 ± 0.2103 (−13.7411%)

Other quantities reported here include data mismatch during history matching, and the value of barrier function (in the form of mean \pm STD), with respect to both the initial and final ensembles (and as in the tables of the previous case study, we also include the relative changes of these quantities in parentheses). For RMSE, the values are calculated with respect to PERMX, PERMY, PERMZ (in the scale of natural logarithm), PORO, and the combination of all these variables, respectively

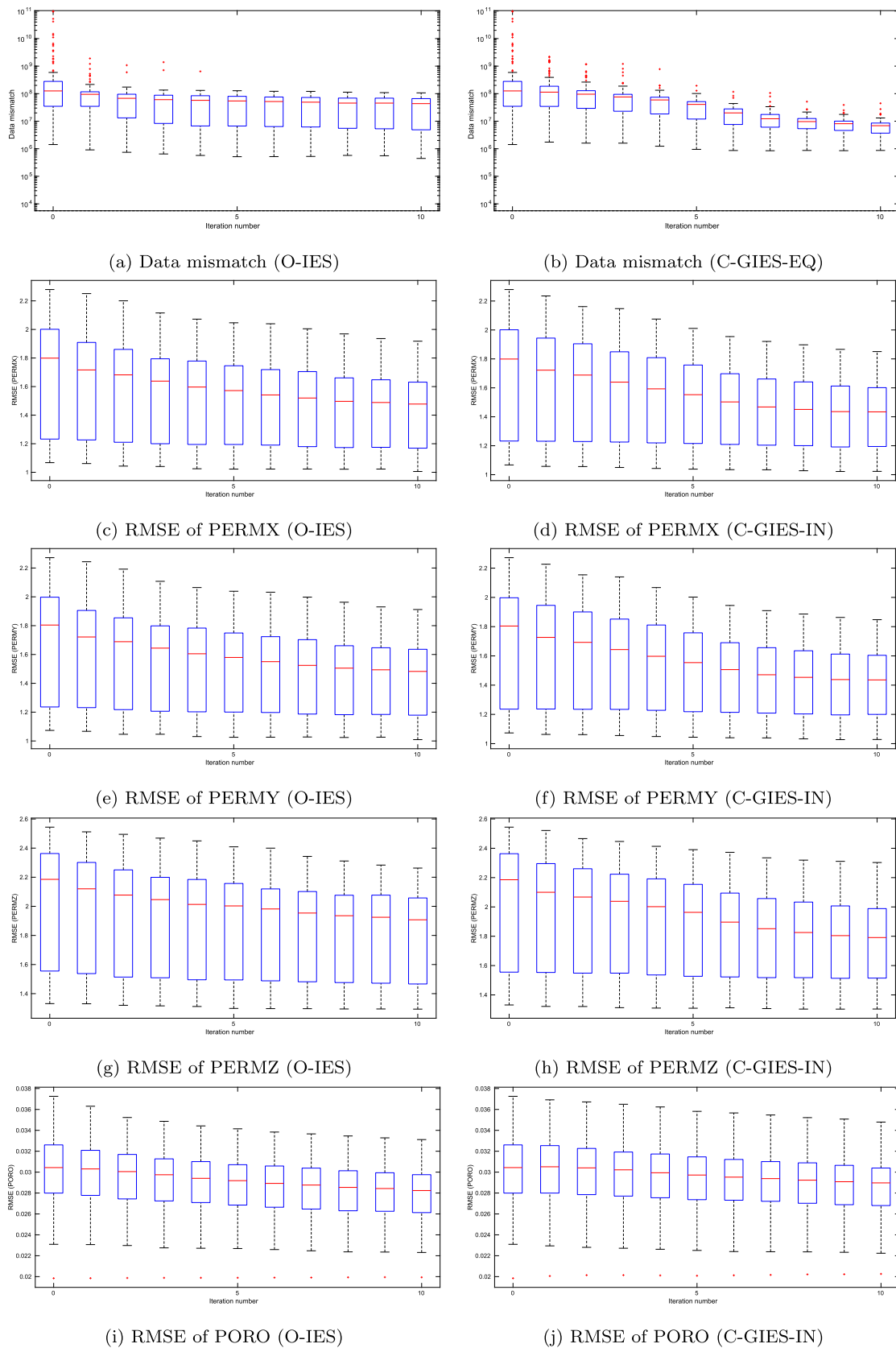


Fig. 7 Box plots of data mismatch and RMSE values at different iteration steps of the O-IES (left column) and the C-GIES-IN (right column) algorithms, in the Brugge benchmark case study

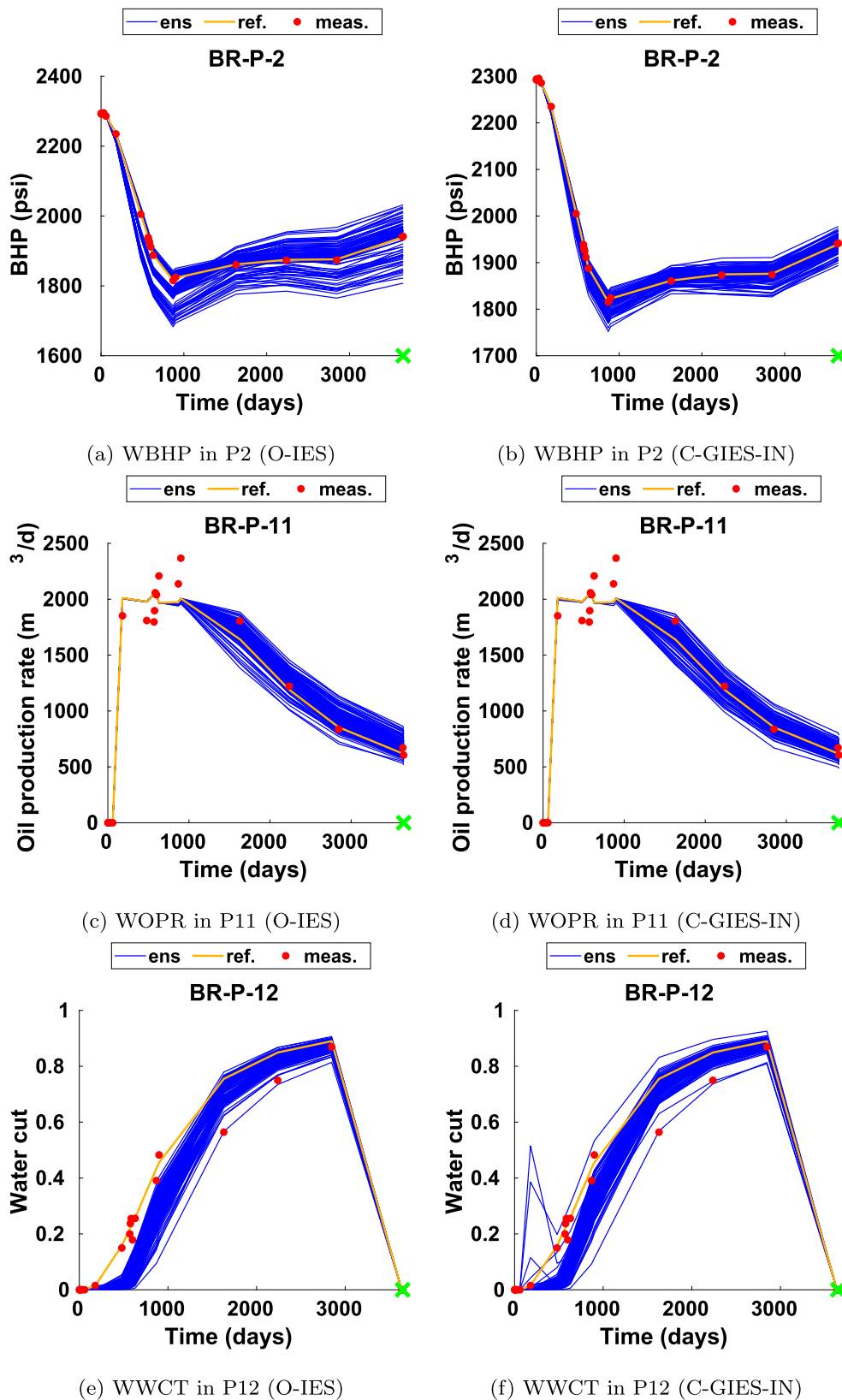


Fig. 8 Profiles of WBHP in P2, WOPR in P11, and WWCT in P12 in the 3D case study, with respect to the O-IES (left column) and C-GIES-IN (right column) algorithms, respectively. In each sub-figure, the orange curve stands for the production data generated by the

reference reservoir model without observation errors; the red dots for noisy observations used in history matching; the blue curves for the simulated production data using the ensembles obtained at the last iteration steps of the O-IES and C-GIES-IN algorithms

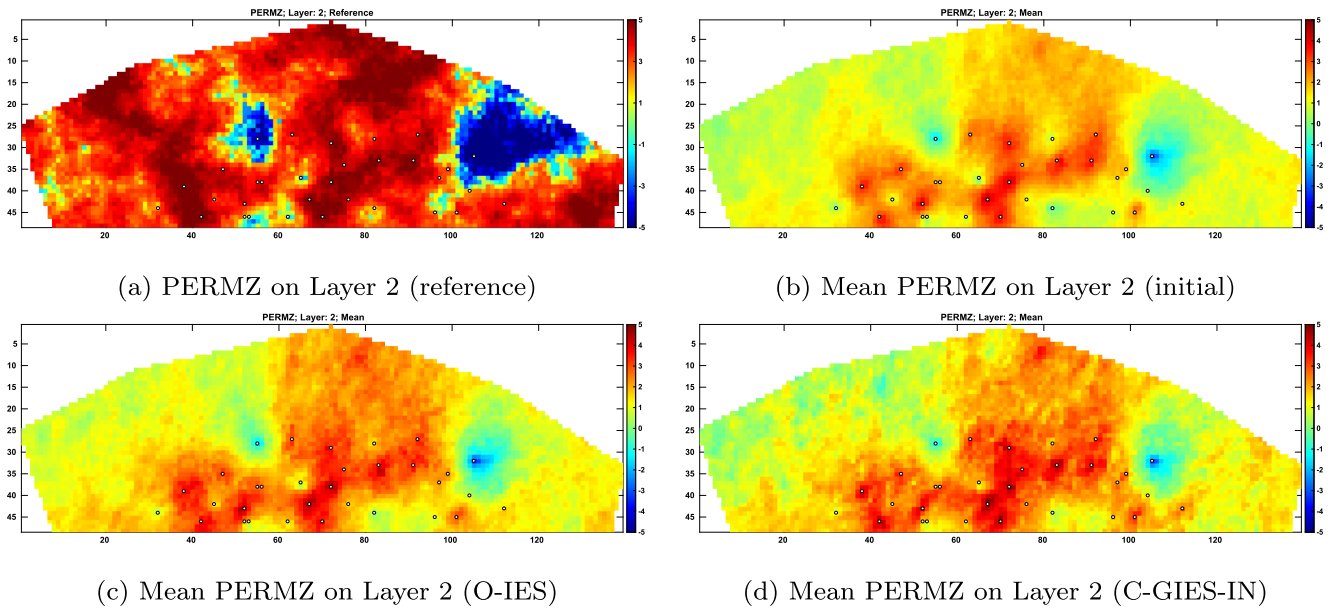


Fig. 9 Reference PERMZ map on Layer 2 of the reservoir model, and mean PERMZ maps with respect to the initial ensemble, and the final ensembles obtained by the O-IES and C-GIES-IN algorithms

algorithms is 10, but these algorithms may stop earlier if the change of the average data mismatch values during two consecutive iteration steps is less than 0.01%.

3.2.2 Numerical results of the 3D case study

Table 3 examines the values of data mismatch, barrier function of the IN-constraint system, and RMSE (separate for PERMX, PERMY, PERMZ, PORO, and also the combination of all model variables), with respect to both

the initial ensemble, and the final ensembles obtained by the O-IES and C-GIES-IN algorithms. All these quantities are reported in the form of mean \pm STD. In comparison to the values of data mismatch, barrier function and RMSE of the initial ensemble, those of the final ensembles tend to be lower, indicating that the performance is improved by using either history-matching algorithm, and that in this particular case study, it is possible to simultaneously reduce both data mismatch and barrier function values during history matching. In comparison to the O-IES algorithm,

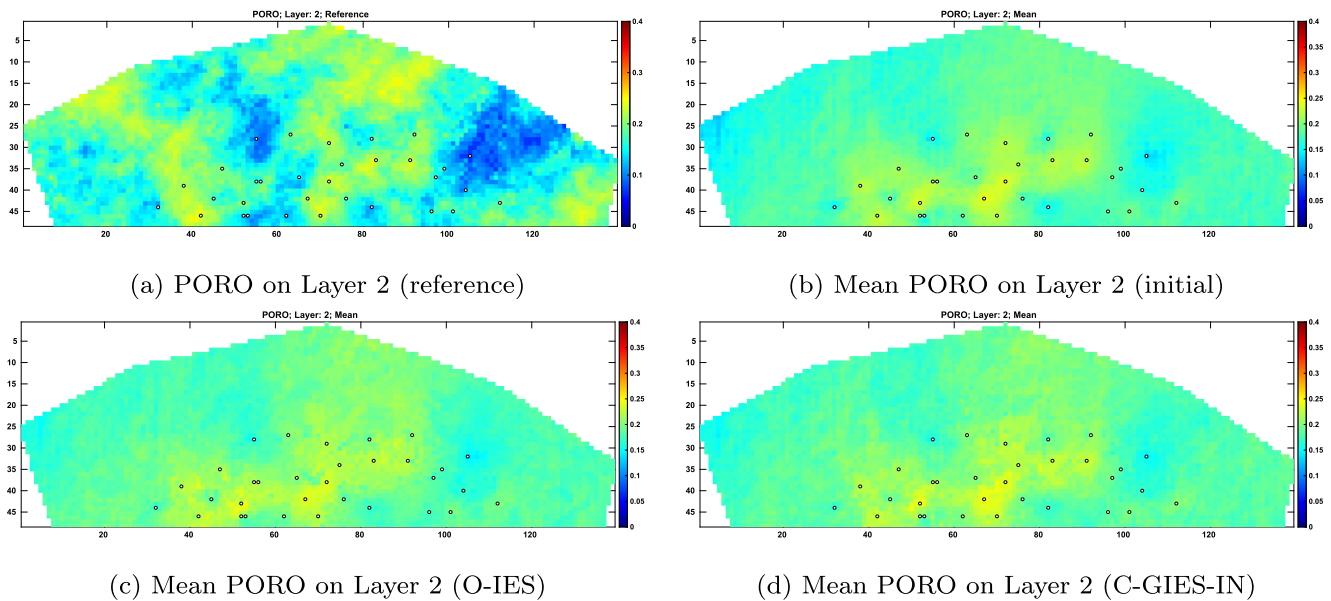


Fig. 10 As in Fig. 9, but for PORO maps on Layer 2

the C-GIES-IN algorithm tends to result in lower data mismatch and barrier function values (meaning that the box constraints are better honored). In terms of RMSE, the C-GIES-IN algorithm leads to lower (average) estimation errors for PERMX, PERMY and PERMZ, but slightly higher errors for the estimation of PORO. The overall RMSE (by putting all model variables together) induced by the C-GIES-IN algorithm also tends to be smaller than that of the O-IES algorithm.

Next, Fig. 7 shows the box plots of data mismatch and RMSE values of PERMX, PERMY, PERMZ and PORO at different iteration steps of the O-IES and C-GIES-IN algorithms. In both history-matching algorithms, all the examined quantities tend to decrease as the iteration step increases. As can be expected from the results in Table 3, in comparison to the O-IES algorithm, the C-GIES-IN algorithm leads to lower data mismatch and RMSEs of the estimated permeability values, but results in slightly higher errors in the estimated porosity values.

Moreover, Fig. 8 reports the profiles of some production data, including WBHP in P2, WOPR in P11, and WWCT in P12. As can be seen there, in general, the forecast production data of the C-GIES-IN algorithm (right column) tend to match the observed data better than the O-IES algorithm (left column) does.

For an inspection on the reservoir models, Figs. 9 and 10 show the reference and mean PERMZ and PORO maps, respectively, on Layer 2. In comparison to the mean PERMZ map of the initial ensemble, more substantial differences can be found in the mean PERMZ map of the final ensemble, obtained by either the O-IES or the C-GIES-IN algorithm. In contrast, the differences among the mean PORO maps of the initial and final ensembles appear less noticeable. The possible reason behind this observation is that in this particular case study, the changes of permeability values are more sensitive to production data than the changes of porosity values, as also noticed in the previous work [26]. On the other hand, by comparing the final mean PERMZ and PORO maps to those of the reference model, one can see that the resemblances of the final estimations to the reference maps may not be satisfactory. One can improve the estimation quality by introducing additional sources of information, e.g., 4D seismic data, into history matching (see, for example, [27, 28]), which, however, is a topic beyond the focus of the current work.

4 Discussion and conclusion

This work presents an ensemble-based workflow to handle data assimilation with soft constraints (DASC). The main idea here is to treat both equality- and inequality-constraint

systems as some perfect observation operators, and then augment these constraint and the original observation systems. Under this setting, DASC aims to simultaneously assimilate multiple sources of information into model variables. From the optimization-theoretic perspective, DASC can be recast as a generalized minimum-average-cost (GMAC) problem, in which the cost function contains a data mismatch term that consists of three discrepancy sub-terms, induced by the original observation system, the equality-constraint system, and the inequality-constraint system, respectively. The generic GMAC problem can be approximately solved by the generalized iterative ensemble smoother (GIES), and in the context of constrained data assimilation, it leads to a constrained GIES (C-GIES) algorithm, where we choose the square of a weighted ℓ_2 norm in the regularization term of the cost function. Although not investigated in the current work, in principle one can develop infinitely many C-GIES algorithms by adopting a different regularization term, following the example of the class of ℓ_p^q -GIES algorithms in [23]; or in a similar way, by changing the distance metrics \mathcal{D}_d , \mathcal{D}_{eq} and \mathcal{D}_{in} or the transform operators \mathcal{T}_d , \mathcal{T}_{eq} and \mathcal{T}_{in} in the data mismatch term (cf. Eq. 28).

The presence of constraints leads to two potential issues in practice. One is regarding the possible high dimensionality induced by the constraints, and the other is about localization for the constraint systems. To tackle the first issue, we propose to choose the distance metrics \mathcal{D}_{eq} and \mathcal{D}_{in} in such a way that the resulting Hessians are diagonal matrices. To deal with the second issue, we apply correlation-based adaptive localization to the terms related to the projected gradients, which leads to some tapering matrices in the dimension of model size by ensemble size, and is useful for handling a large amount of “effective” data, from either the original observation system, or the constraint systems, or both.

We then apply the proposed C-GIES algorithm to one 2D and one 3D case studies. The numerical results indicate that properly incorporating the constraints does help improve the performance of data assimilation. In particular, the experiment settings of the 3D example are close to those of a real field case study, and we have observed that the proposed C-GIES algorithm is able to efficiently handle both the issues of high dimensionality induced by the constraint systems, and localization for the constraint systems, while achieving reasonably good performance. This implies that the proposed C-GIES algorithm has the potential to be used in real field case studies.

Finally, we bring up two remaining open problems in this line of research. One problem is how to optimally allocate relative weights to the discrepancy sub-terms with respect to the original observation system and the constraint systems,

respectively; and the other is that, at a higher level, how to choose an optimal cost function to deduce a suitable GIES or C-GIES algorithm for a given data assimilation problem.

Acknowledgements The authors would like to thank two anonymous reviewers for their valuable and constructive comments and suggestions. XL acknowledges financial support from the Research Council of Norway (RCN) through the Petromaks-2 project DIGIRES (RCN no. 280473) and the industrial partners AkerBP, Wintershall DEA, Vår Energi, Petrobras, Equinor, Lundin and Neptune Energy; WC acknowledges financial support from the National IOR centre of Norway (RCN no. 230303), which is funded by the RCN and industry partners ConocoPhillips, Aker BP, Vår Energi, Equinor, Neptune Energy, Lundin, Halliburton, Schlumberger, and Wintershall Dea. We would also like to thank Schlumberger for providing academic licenses to ECLIPSE®

Funding Open Access funding provided by NORCE Norwegian Research Centre AS.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Albers, D.J., Blancquart, P.A., Levine, M.E., Seylabi, E.E., Stuart, A.: Ensemble kalman methods with constraints. *Inverse Probl.* **35**(9), 095007 (2019)
- Amor, N., Rasool, G., Bouaynaya, N.C.: Constrained state estimation—a review. [arXiv:1807.03463](https://arxiv.org/abs/1807.03463) (2018)
- Brembeck, J.: Nonlinear constrained moving horizon estimation applied to vehicle position estimation. *Sensors* **19**(10), 2276 (2019)
- Burgers, G., van Leeuwen, P.J., Evensen, G.: On the analysis scheme in the ensemble Kalman filter. *Mon. Wea. Rev.* **126**, 1719–1724 (1998)
- Canchumuni, S.W., Emerick, A.A., Pacheco, M.A.C.: History matching geological facies models based on ensemble smoother and deep generative models. *J. Pet. Sci. Eng.* **177**, 941–958 (2019)
- Chada, N.K., Schillings, C., Weissmann, S.: On the incorporation of box-constraints for ensemble Kalman inversion. *Found. Data Sci.* **1**(4), 433 (2019)
- Chen, Y., Oliver, D.S.: Cross-covariances and localization for enKF in multiphase flow data assimilation. *Comput. Geosci.* **14**, 579–601 (2010)
- Chen, Y., Oliver, D.: Levenberg-Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification. *Comput. Geosci.* **17**, 689–703 (2013)
- De Geeter, J., Van Brussel, H., De Schutter, J., Decréton, M.: A smoothly constrained kalman filter. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(10), 1171–1177 (1997)
- Emerick, A.A., Reynolds, A.C.: History matching a field case using the ensemble Kalman filter with covariance localization. *SPE Reserv. Eval. Eng.* **14**, 423–432 (2011). SPE-141216-PA
- Emerick, A.A., Reynolds, A.C.: Ensemble smoother with multiple data assimilation. *Comput. Geosci.* **55**, 3–15 (2012)
- Evensen, G., Raanes, P.N., Stordal, A.S., Hove, J.: Efficient implementation of an iterative ensemble smoother for big-data assimilation and reservoir history matching. *Front. Appl. Math. Stat.* **5**, 47 (2019)
- Forouzanfar, F., Wu, X.H.: Constrained iterative ensemble smoother for multi solution search assisted history matching. *Comput. Geosci.* **25**, 1593–1604 (2021)
- Gupta, N., Hauser, R.: Kalman filtering with equality and inequality state constraints. [arXiv:0709.2791](https://arxiv.org/abs/0709.2791) (2007)
- Herty, M., Visconti, G.: Continuous limits for constrained ensemble kalman filter. *Inverse Probl.* **36**(7), 075006 (2020)
- Iglesias, M.A., Law, K.J., Stuart, A.M.: Ensemble Kalman methods for inverse problems. *Inverse Probl.* **29**(4), 045001 (2013)
- Jafarpour, B.: Wavelet reconstruction of geologic facies from nonlinear dynamic flow measurements. *IEEE Trans. Geosci. Remote Sens.* **49**, 1520–1535 (2011)
- Janjić, T., McLaughlin, D., Cohn, S.E., Verlaan, M.: Conservation of mass and preservation of positivity with ensemble-type kalman filter algorithms. *Mon. Weather. Rev.* **142**(2), 755–773 (2014)
- Khaki, M., Ait-El-Fquih, B., Hoteit, I., Forootan, E., Awange, J., Kuhn, M.: A two-update ensemble Kalman filter for land hydrological data assimilation with an uncertain constraint. *J. Hydrol.* **555**, 447–462 (2017)
- Lauvernet, C., Brankart, J.M., Castruccio, F., Broquet, G., Brasseur, P., Verron, J.: A truncated Gaussian filter for data assimilation with inequality constraints: application to the hydrostatic stability condition in ocean models. *Ocean Model.* **27**(1–2), 1–17 (2009)
- Lorentzen, R., Flornes, K., Nævdal, G.: History matching channelized reservoirs using the ensemble Kalman filter. *SPE J.* **17**, 137–151 (2012)
- Lorentzen, R., Luo, X., Bhakta, T., Valestrand, R.: History matching the full Norne field model using seismic and production data. *SPE J.* **24**, 1452–1467 (2019). SPE-194205-PA
- Luo, X.: Novel iterative ensemble smoothers derived from a class of generalized cost functions. *Comput. Geosci.* **25**, 1159–1189 (2021)
- Luo, X., Bhakta, T.: Automatic and adaptive localization for ensemble-based history matching. *J. Pet. Sci. Eng.* **184**, 106559 (2020)
- Luo, X., Stordal, A., Lorentzen, R., Nævdal, G.: Iterative ensemble smoother as an approximate solution to a regularized minimum-average-cost problem: theory and applications. *SPE J.* **20**, 962–982 (2015). <https://doi.org/10.2118/176023-PA>. SPE-176023-PA
- Luo, X., Bhakta, T., Jakobsen, M., Nævdal, G.: An ensemble 4D-seismic history-matching framework with sparse representation based on wavelet multiresolution analysis. *SPE J.* **22**, 985–1010 (2017). <https://doi.org/10.2118/180025-PA>. SPE-180025-PA
- Luo, X., Bhakta, T., Jakobsen, M., Nævdal, G.: Efficient big data assimilation through sparse representation: a 3D benchmark case study in petroleum engineering. *PLOS ONE* **13**, e0198586 (2018)
- Luo, X., Bhakta, T., Nævdal, G.: Correlation-based adaptive localization with applications to ensemble-based 4D seismic history matching. *SPE J.* **23**, 396–427 (2018). <https://doi.org/10.2118/185936-PA>. SPE-185936-PA
- Luo, X., Lorentzen, R.J., Valestrand, R., Evensen, G.: Correlation-based adaptive localization for ensemble-based history matching: applied to the Norne field case study. *SPE Reserv. Eval. Eng.* **22**, 1084–1109 (2019). <https://doi.org/10.2118/191305-PA>. SPE-191305-PA

30. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, Berlin (2006)
31. Peters, L., Arts, R., Brouwer, G., Geel, C., Cullick, S., Lorentzen, R.J., Chen, Y., Dunlop, N., Vossepoel, F.C., Xu, R., et al: Results of the Brugge benchmark study for flooding optimization and history matching. *SPE Reserv. Eval. Eng.* **13**, 391–405 (2010)
32. Phale, H.A., Oliver, D.S.: Data assimilation using the constrained ensemble kalman filter. *SPE J.* **16**(02), 331–342 (2011)
33. Prakash, J., Patwardhan, S.C., Shah, S.L.: Constrained nonlinear state estimation using ensemble kalman filters. *Ind. Eng. Chem. Res.* **49**(5), 2242–2253 (2010)
34. Simon, D.: Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory Appl.* **4**(8), 1303–1318 (2010)
35. Simon, D., Chia, T.L.: Kalman filtering with state equality constraints. *IEEE Trans. Aerosp. Electron. Syst.* **38**(1), 128–136 (2002)
36. Soares, R., Luo, X., Evensen, G., Bhakta, T.: 4D seismic history matching: Assessing the use of a dictionary learning based sparse representation method. *J. Pet. Sci. Eng.* **195**, 107763 (2020)
37. Soares, R.V., Luo, X., Evensen, G., Bhakta, T.: Handling big models and big data sets in history-matching problems through an adaptive local analysis scheme. *SPE J.* **26**(02), 973–992 (2021)
38. Strebelle, S.: Conditional simulation of complex geological structures using multiple-point statistics. *Math. Geol.* **34**, 1–21 (2002)
39. Teixeira, B.O., Tôrres, L.A., Aguirre, L.A., Bernstein, D.S.: Unscented filtering for interval-constrained nonlinear systems. In: 2008 47th IEEE Conference on Decision and Control, pp. 5116–5121 (2008)
40. Thacker, W.: Data assimilation with inequality constraints. *Ocean Model.* **16**(3–4), 264–276 (2007)
41. Vachhani, P., Narasimhan, S., Rengaswamy, R.: Robust and reliable estimation via unscented recursive nonlinear dynamic data reconciliation. *J. Process Control* **16**(10), 1075–1086 (2006)
42. Yang, C., Blasch, E.: Kalman filtering with nonlinear state constraints. *IEEE Trans. Aerosp. Electron. Syst.* **45**(1), 70–84 (2009)
43. Yu, M., Chen, W.H., Chambers, J.: Truncated unscented particle filter for dealing with non-linear inequality constraints. In: 2014 Sensor Signal Processing for Defence (SSPD), pp. 1–5. IEEE (2014)
44. Zhang, X.L., Michelén-Ströfer, C., Xiao, H.: Regularized ensemble Kalman methods for inverse problems. *J. Comput. Phys.* **416**, 109517 (2020)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.