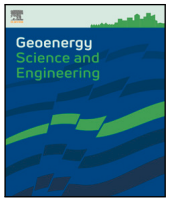
231 (2023) 212404

Contents lists available at ScienceDirect

Geoenergy Science and Engineering

journal homepage: www.sciencedirect.com/journal/geoenergy-science-and-engineering



Hyper-parameter optimization for improving the performance of localization in an iterative ensemble smoother

Xiaodong Luo [a,*], William C. Cruz [b], Xin-Lei Zhang [c,d], Heng Xiao [e]

[a] Norwegian Research Centre (NORCE), Nygårdsgaten 112, Bergen, 5008, Norway
[b] University of Stavanger, Kjell Arholms gate 41, Stavanger, 4021, Norway
[c] The State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing, PR China
[d] School of Engineering Sciences, University of Chinese Academy of Sciences, Beijing, PR China
[e] Stuttgart Center for Simulation Science (SC SimTech), University of Stuttgart, Stuttgart Germany

ARTICLE INFO

Keywords:
Ensemble data assimilation
Iterative ensemble smoother (IES)
Automatic and adaptive localization (AutoAdaLoc)
Parameterized localization
Continuous hyper-parameter OPtimization (CHOP)

ABSTRACT

This work aims to help improve the performance of an iterative ensemble smoother (IES) in reservoir data assimilation problems, by introducing a data-driven procedure to optimize the choice of certain algorithmic hyper-parameters in the IES. Generally speaking, algorithmic hyper-parameters exist in various data assimilation algorithms. Taking IES as an example, localization is often useful for improving its performance, yet applying localization to an IES also introduces a certain number of algorithmic hyper-parameters, such as localization length scales, in the course of data assimilation. While different methods have been developed in the literature to address the problem of properly choosing localization length scales in various circumstances, many of them are tailored to specific problems under consideration, and may be difficult to directly extend to other problems. In addition, conventional hyper-parameter tuning methods determine the values of localization length scales based on either empirical (e.g., using experience, domain knowledge, or simply the practice of trial and error) or analytic (e.g., through statistical analyses) rules, but few of them use the information of observations to optimize the choice of hyper-parameters. The current work proposes a generic, data-driven hyper-parameter tuning strategy that has the potential to overcome the aforementioned issues. With this proposed strategy, hyper-parameter optimization is converted into a conventional parameter estimation problem, in such a way that observations are utilized to guide the choice of hyper-parameters. One noticeable feature of the proposed hyper-parameter tuning strategy is that it iteratively estimates an ensemble of hyper-parameters. In doing so, the resulting hyper-parameter tuning procedure receives some practical benefits inherent to conventional ensemble data assimilation algorithms, including the nature of being derivative-free, the ability to provide uncertainty quantification to some extent, and the capacity to handle a large number of hyper-parameters. Through 2D and 3D case studies, it is shown that when the proposed hyper-parameter tuning strategy is applied to tune a set of localization length scales (up to the order of 10^3) in a parameterized localization scheme, superior data assimilation performance is obtained in comparison to an alternative hyper-parameter tuning strategy without utilizing the information of observations.

1. Introduction

Reservoir data assimilation (also known as history matching) aims to find one or multiple set(s) of reservoir model parameters to match the available field data to a good extent. Among others, ensemble-based history-matching methods (Aanonsen et al., 2009; Oliver and Chen, 2010), such as ensemble Kalman filter (EnKF, see Evensen, 2009; Nævdal et al., 2005), ensemble smoother (ES, see Skjervheim and Evensen, 2011; Van Leeuwen and Evensen, 1996) and their iterative versions, e.g., (Chen and Oliver, 2013; Emerick and Reynolds, 2012; Luo et al., 2015; Luo, 2021; Evensen et al., 2019), have been extensively used to tackle reservoir characterization problems. In real applications of ensemble-based methods, small ensembles are often adopted to reduce the computational costs of reservoir simulations. However, using a small ensemble size often leads to rank deficiency and sampling errors in ensemble methods (Anderson, 2012; Hamill et al., 2009),

* Corresponding author.
E-mail addresses: xluo@norceresearch.no (X. Luo), williamchalub@gmail.com (W.C. Cruz), zhangxinlei@imech.ac.cn (X.-L. Zhang).

https://doi.org/10.1016/j.geoen.2023.212404
Received 2 January 2023; Received in revised form 1 June 2023; Accepted 10 October 2023
Available online 14 October 2023

and inferior performance could thus be obtained in data assimilation problems.

To alleviate the adversarial effects of the small ensemble size, a common practice in the community is to equip an ensemble data assimilation algorithm with a localization scheme, which in effect modifies the model update formula in the ensemble data assimilation algorithm. Various localization techniques have been developed from different perspectives, and been applied for different purposes (Anderson, 2012; Anderson and Lei, 2013; Arroyo et al., 2008; Bishop and Hodyss, 2007; Chen and Oliver, 2010, 2017; Emerick and Reynolds, 2011; Fertig et al., 2007; Hamill et al., 2001; Lacerda et al., 2019; Luo et al., 2018; Luo and Bhakta, 2020; Ranazzi et al., 2022; Soares et al., 2021).

For reservoir data assimilation problems, it is customary to conduct Kalman-gain localization, in which a tapering matrix is applied to modify the Kalman-gain-type matrix, in the sense that the original Kalman-gain-type matrix is replaced by the Schur product between it and the tapering matrix. Previously, Kalman-gain localization in reservoir applications has been largely focused on using distances between the physical locations of model variables and observations to calculate the tapering coefficients (Chen and Oliver, 2010, 2017; Emerick and Reynolds, 2011), which is referred to as distance-based localization hereafter. A number of issues, however, have been noticed in distance-based localization, which includes the dependence on the availability of physical locations of both model variables and observations, the difficulty in dealing with non-local observations, the non-adaptivity to some factors like the temporal change of system dynamics, ensemble size, different types of model variables and/or observations, and case-dependent implementation (i.e., the transferability of the localization scheme among different case studies) (Luo et al., 2019).

In light of these challenges, there have been some recent efforts (Anderson, 2012; Furrer and Bengtsson, 2007; Lacerda et al., 2019; Luo et al., 2018; Luo and Bhakta, 2020; Ranazzi et al., 2022; Soares et al., 2021) dedicated to tackling the above-mentioned issues, which use sample correlations between model variables and simulated observations (or between model variables themselves), rather than the distances between their physical locations, to compute the tapering coefficients, and which are thus referred to as correlation-based localization hereafter. For reservoir data assimilation problems, it has been shown that correlation-based localization is able to avoid or mitigate the aforementioned issues (Luo et al., 2019), works reasonably well in both synthetic (Lacerda et al., 2019; Luo et al., 2018; Luo and Bhakta, 2020; Ranazzi et al., 2022; Soares et al., 2021) and real field (Luo et al., 2019; Lorentzen et al., 2020) case studies.

Most of the existing localization schemes, being either distance- or correlation-based, have some inherent hyper-parameters that need to be determined in one way or another. The type of the hyper-parameters could be localization length scale (Chen and Oliver, 2010; Emerick and Reynolds, 2011; Hamill et al., 2001; Luo and Bhakta, 2020; Soares et al., 2021), localization radius (Chen and Oliver, 2017), certain threshold values (Lacerda et al., 2019; Luo et al., 2018), or coefficients inside the tapering functions (Anderson, 2012; Bishop and Hodyss, 2007; Ranazzi et al., 2022).

In many cases, the hyper-parameters of localization schemes are determined based on either experience, expertise, domain knowledge or simply trial and error. While this *empirical tuning strategy* may still lead to satisfactory data assimilation performance, it could be time-consuming to search for the optimal values, cumbersome to handle multiple hyper-parameters, and difficult to generalize to other case studies or disciplines (for reflection, one may draw similarity of this tuning strategy to manual history matching). A theoretically more appealing strategy is to carry out a certain statistical analysis to determine the values of hyper-parameters, as was done in, e.g., Furrer and Bengtsson (2007), Luo and Bhakta (2020). Often, the statistical analysis is derived based on some simplifying assumptions (e.g., by assuming sampling errors in sample correlations follow certain Gaussian distributions as in Luo and Bhakta, 2020). Moreover, once specified (prior to data assimilation), the chosen hyper-parameter values are fixed, and do not change in the process of data assimilation hereafter. For this reason, we refer to this hyper-parameter tuning strategy as the *prior tuning strategy*.

The focus of the current work is on further improving the performance of a correlation-based automatic and adaptive localization (AutoAdaLoc) scheme in Luo and Bhakta (2020), by introducing a *posterior tuning strategy* to continuously update multiple hyper-parameters (localization length scales) of a parameterized AutoAdaLoc (P-AutoAdaLoc) scheme in the course of data assimilation. This posterior tuning strategy is built upon a recently proposed Continuous Hyper-parameter OPtimization (CHOP) procedure (Luo and Xia, 2022), in which a model update algorithm (either from an ensemble-based method or not) is treated as a parametric mapping that is parameterized by some algorithmic hyper-parameters (e.g., length scales of a localization scheme). From this perspective, these algorithmic hyper-parameters have an influence on the updated reservoir model, hence the corresponding set of simulated observations. Therefore, similar to the criterion used in Luo et al. (2015) to estimate reservoir model parameters, one can update these hyper-parameters in such a way that the updated hyper-parameters lead to reduced data mismatch between real and simulated observations. In other words, in the posterior tuning strategy, one can essentially convert hyper-parameter optimization into a conventional parameter estimation problem.

One implication from the above discussion is that the hyper-parameter optimization problem can be solved by a relevant data assimilation method. For this purpose, we adopt the iterative ensemble smoother (IES) proposed by Luo et al. (2015) in the current work, although other algorithms (whether ensemble-based or not) can also be used. The main reason behind this choice is that we use the same IES algorithm to estimate reservoir model parameters. As such, hyper-parameter optimization through the same IES algorithm entails a natural extension of our ensemble-based data assimilation workflow, which can now be used to iteratively update both ordinary reservoir model parameters and algorithmic hyper-parameters (cf. Algorithm 2 later). Under this setting, the CHOP procedure is also ensemble-based, hence it possesses the same practical benefits inherent to ensemble-based algorithms, including the nature of being non-intrusive and derivative-free, the ability of providing uncertainty quantification to some extent, and the capacity of handling a relatively large amount of hyper-parameters.

In terms of novelty, to the best of our knowledge, it is the first time that an ensemble-based posterior tuning procedure is applied to optimize algorithmic hyper-parameters in reservoir data assimilation problems. In addition, in the previous work (Luo and Xia, 2022), one only updates algorithmic hyper-parameters through the CHOP procedure, while keeping model variables (model state and/or parameters) fixed. In contrast, in the current work, the proposed ensemble data assimilation workflow aims to update both reservoir model parameters and algorithmic hyper-parameters, which requires the introduction of an alternating optimization procedure, as will be elaborated later.

In the sequel, we first provide an overview of the ensemble-based workflow used in the current work, which contains the IES algorithm of Luo et al. (2015) as the data assimilation method, and the AutoAdaLoc scheme of Luo and Bhakta (2020) for Kalman-gain localization. We then propose the P-AutoAdaLoc scheme that can take multiple localization length scales as the algorithmic hyper-parameters. To facilitate the replacement of the original AutoAdaLoc scheme by the P-AutoAdaLoc scheme, we introduce an ensemble-based CHOP procedure that can iteratively update an ensemble of algorithmic hyper-parameters through the same IES algorithm. After presenting this methodological development, we proceed to compare in three case studies the performance of the IES-based data assimilation workflows that adopt the original AutoAdaLoc and the proposed P-AutoAdaLoc schemes, respectively, for localization. Finally, we conclude the whole work with some technical discussions and possible future research directions.

2. Methodology

2.1. Iterative ensemble smoother (IES) for reservoir data assimilation problems

Let $\mathbf{m} \in \mathbb{R}^m$ be an m-dimensional vector containing a set of parameters to be estimated by a certain data assimilation algorithm, which is referred to as a reservoir model hereafter. In addition, let $\mathbf{g} : \mathbb{R}^m \to \mathbb{R}^p$ stand for the forward reservoir simulator that maps a reservoir model $\mathbf{m}$ to a p-dimensional vector $\mathbf{d}^{\text{sim}} \equiv \mathbf{g}(\mathbf{m}) \in \mathbb{R}^p$ (called simulated observations), and $\mathbf{d}^o \in \mathbb{R}^p$ for the p-dimensional real observations. We assume that $\mathbf{d}^o$ is contaminated by certain observation errors that follow a multivariate Gaussian distribution with zero mean and covariance $\mathbf{C}_d \in \mathbb{R}^{p\times p}$.

An IES consists of a number of iteration steps, and at each step the IES updates an ensemble of reservoir models to a new ensemble, following a certain rule (Emerick and Reynolds, 2012; Chen and Oliver, 2013; Luo et al., 2015; Luo, 2021; Evensen et al., 2019). In the current work, the IES algorithm developed by Luo et al. (2015), referred to as regularized Levenberg–Marquardt algorithm for a minimum-average-cost problem (RLM-MAC), is adopted as the data assimilation algorithm, although it is expected that the main idea behind the new localization scheme can be naturally extended to other IES algorithms. In this subsection, we provide a brief description of the RLM-MAC algorithm, which is also referred to as the IES algorithm for brevity.

Let $\mathbf{M}^i \equiv \{\mathbf{m}_j^i : \mathbf{m}_j^i \in \mathbb{R}^m\}_{j=1}^{N_e}$ be an ensemble of N_e reservoir models obtained at the ith iteration step. The RLM-MAC algorithm updates $\mathbf{M}^i$ to a new ensemble $\mathbf{M}^{i+1} \equiv \{\mathbf{m}_j^{i+1} : \mathbf{m}_j^{i+1} \in \mathbb{R}^m\}_{j=1}^{N_e}$ by solving the following minimum-average-cost (MAC) problem:

$$\min_{\{\mathbf{m}_j^{i+1}\}_{j=1}^{N_e}} \frac{1}{N_e}\sum_{j=1}^{N_e} C_j^{i+1}\,; \tag{1}$$

$$C_j^{i+1} \equiv \left(\mathbf{d}_j^o - \mathbf{g}\left(\mathbf{m}_j^{i+1}\right)\right)^T \mathbf{C}_d^{-1}\left(\mathbf{d}_j^o - \mathbf{g}\left(\mathbf{m}_j^{i+1}\right)\right) + \gamma^i\left(\mathbf{m}_j^{i+1} - \mathbf{m}_j^i\right)^T \left(\mathbf{C}_m^i\right)^{-1}\left(\mathbf{m}_j^{i+1} - \mathbf{m}_j^i\right), \tag{2}$$

at each iteration step. In Eq. (2), $\mathbf{d}_j^o$ ($j = 1, 2, \cdots, N_e$) represent perturbations of $\mathbf{d}^o$, which are samples drawn from the multivariate Gaussian distribution $N(\mathbf{d}^o, \mathbf{C}_d)$; γ^i is a scalar coefficient (regularization parameter) whose choice follows the rules in Luo et al. (2015), and $\mathbf{C}_m^i \equiv \mathbf{S}_m^i\left(\mathbf{S}_m^i\right)^T$ corresponds to the sample covariance matrix with respect to the ensemble $\mathbf{M}^i$.

It is shown in Luo et al. (2015) that the MAC problem (Eq. (2)) is approximately solved by the following model update formula (the RLM-MAC algorithm):

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \mathbf{S}_m^i(\mathbf{S}_g^i)^T\left(\mathbf{S}_g^i(\mathbf{S}_g^i)^T + \gamma^i\,\mathbf{C}_d\right)^{-1} \times \left(\mathbf{d}_j^o - \mathbf{g}\left(\mathbf{m}_j^i\right)\right), \text{ for } j = 1, 2, \cdots, N_e\,; \tag{3}$$

$$\mathbf{S}_m^i = \frac{1}{\sqrt{N_e - 1}}\left[\mathbf{m}_1^i - \overline{\mathbf{m}}^i, \cdots, \mathbf{m}_{N_e}^i - \overline{\mathbf{m}}^i\right]\,; \quad \overline{\mathbf{m}}^i = \frac{1}{N_e}\sum_{j=1}^{N_e}\mathbf{m}_j^i\,; \tag{4}$$

$$\mathbf{S}_g^i = \frac{1}{\sqrt{N_e - 1}}\left[\mathbf{g}(\mathbf{m}_1^i) - \mathbf{g}(\overline{\mathbf{m}}^i), \cdots, \mathbf{g}(\mathbf{m}_{N_e}^i) - \mathbf{g}(\overline{\mathbf{m}}^i)\right]. \tag{5}$$

In a practical implementation, some modification of the update formula, Eq. (3), may be introduced. For instance, Eq. (3) can be re-written as

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \mathbf{S}_m^i(\tilde{\mathbf{S}}_g^i)^T\left(\tilde{\mathbf{S}}_g^i(\tilde{\mathbf{S}}_g^i)^T + \gamma^i\,\mathbf{I}_p\right)^{-1}\left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}}\left(\mathbf{m}_j^i\right)\right)\,; \tag{6}$$

$$\tilde{\mathbf{S}}_g^i \equiv \mathbf{C}_d^{-1/2}\mathbf{S}_g^i\,; \tag{7}$$

$$\tilde{\mathbf{d}}_j^o \equiv \mathbf{C}_d^{-1/2}\mathbf{d}_j^o\,; \tag{8}$$

$$\tilde{\mathbf{g}}\left(\mathbf{m}_j^i\right) \equiv \mathbf{C}_d^{-1/2}\mathbf{g}\left(\mathbf{m}_j^i\right), \tag{9}$$

where $\mathbf{I}_p$ is the p-dimensional identity matrix. In Eq. (6), one normalizes relevant quantities in the observation space by the square root matrix $\mathbf{C}_d^{-1/2}$ of $\mathbf{C}_d^{-1}$ to avoid potential numerical issues caused by different orders of magnitude in simulated and real observations. In addition, to further improve the numerical stability of the IES, a truncated singular value decomposition (TSVD) can be applied to $\tilde{\mathbf{S}}_g^i$. For brevity, here we skip the technical details of the TSVD-based implementation of the IES and refer readers to Luo et al. (2015) for more information.

2.2. Correlation-based automatic and adaptive localization (AutoAdaLoc) in the IES

When applying an IES to practical reservoir data assimilation problems, one can only afford to run a relatively small ensemble of reservoir models, typically in the order of $\mathcal{O}(10^2)$. One consequence of using a small ensemble is that sampling errors become substantial, and often deteriorate the performance of data assimilation. For performance improvement, localization is often introduced to the IES algorithm.

Without loss of generality, one can re-write the model update formula of an IES as:

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \mathbf{K}^i\left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}}\left(\mathbf{m}_j^i\right)\right), \tag{10}$$

where $\mathbf{K}^i$ is a Kalman-gain-like matrix, e.g., $\mathbf{K}^i = \mathbf{S}_m^i(\tilde{\mathbf{S}}_g^i)^T\left(\tilde{\mathbf{S}}_g^i(\tilde{\mathbf{S}}_g^i)^T + \gamma^i\,\mathbf{I}_p\right)^{-1}$ for Eq. (6).

In reservoir data assimilation problems, localization is conducted by replacing the matrix $\mathbf{K}^i$ by the Schur (Hadamard) product between $\mathbf{K}^i$ and a tapering matrix $\mathbf{T}^i$, so that the model update formula of the IES becomes

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \left(\mathbf{T}^i \circ \mathbf{K}^i\right)\left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}}\left(\mathbf{m}_j^i\right)\right), \tag{11}$$

where $\circ$ is the operator of Schur product.

A localization scheme consists of a set of rules that specify how the tapering matrix $\mathbf{T}^i$ is constructed. In reservoir data assimilation problems, conventional localization schemes use the distances between the physical locations of both reservoir model variables and observation data points to calculate respective tapering matrices (Chen and Oliver, 2010; Emerick and Reynolds, 2011). Physical-distance based localization schemes are often prone to a number of long-standing issues, including, for instance, (1) the need for physical locations of both reservoir model variables and observation data points; (2) the difficulty in handling non-local observations; (3) the lack of adaptivity to certain factors like the ensemble size used in the IES, the change of reservoir fluid dynamics (e.g., due to the effects of field production), the types of model variables and field data, and so on; and (4) case-dependent implementations.

An alternative to distance-based localization is correlation-based localization (Lacerda et al., 2019; Luo et al., 2018; Luo and Bhakta, 2020; Ranazzi et al., 2022), in which a tapering matrix is formulated based on sample correlations – rather than physical distances – between ensembles of model variables and simulated observation data points. As elaborated in Luo et al. (2018, 2019), correlation-based localization is able to overcome or mitigate the aforementioned long-standing issues arising in distance-based localization, and can be used in certain situations, e.g., when model variables and/or observations do not possess physical locations (Luo et al., 2018) or when localization needs to be done for an ensemble of gradients projected onto ensemble subspaces (Luo and Cruz, 2022), where the conventional distance-based localization does not seem to be straightforwardly applicable.

The focus of the current work is to explore the possibility of further improving the performance of a correlation-based automatic and adaptive localization (AutoAdaLoc) scheme proposed in Luo and Bhakta (2020). To this end, we first briefly describe the AutoAdaLoc scheme, and then present a modification of this scheme, with which further performance improvement becomes possible.

The model-update formula with localization, Eq. (11), can be equivalently written in an element-wise way, as follows:

$$m_{j,k}^{i+1} = m_{j,k}^{i} + \sum_{s=1}^{p} \left(t_{k,s}^{i} k_{k,s}^{i} \right) \Delta \tilde{d}_{j,s}^{i}, \tag{12}$$

where $m_{j,k}^{i}$ ($m_{j,k}^{i+1}$) stands for the kth element of $\mathbf{m}_j^i$ ($\mathbf{m}_j^{i+1}$). Likewise, $\Delta \tilde{d}_{j,s}^{i}$ represents the sth element of the innovation term $\Delta \tilde{\mathbf{d}}_j^i \equiv \tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}}\left(\mathbf{m}_j^i\right)$. Meanwhile, $t_{k,s}^{i}$ and $k_{k,s}^{i}$ correspond to the elements on the kth row and the sth column of the matrices $\mathbf{T}^i$ and $\mathbf{K}^i$, respectively.

In the AutoAdaLoc scheme proposed by Luo and Bhakta (2020), the tapering coefficient $t_{k,s}^{i}$ is determined by the following rules:

$$t_{k,s} = f_{GC}\left(\frac{1 - \mathrm{abs}(\rho_{k,s}^{0})}{1 - \theta_{G_k,s}} \right), \text{ for } k = 1, 2, \cdots, m; s = 1, 2, \cdots, p, \tag{13}$$

with

$$f_{GC}(z) = \begin{cases} -\frac{1}{4}z^5 + \frac{1}{2}z^3 + \frac{5}{8}z^3 - \frac{5}{3}z^2 + 1, & \text{if } 0 \le z \le 1; \\ \frac{1}{12}z^5 - \frac{1}{2}z^4 + \frac{5}{8}z^3 + \frac{5}{3}z^2 - 5z + 4 - \frac{2}{3}z^{-1}, & \text{if } 1 < z \le 2; \\ 0, & \text{if } z > 2, \end{cases} \tag{14}$$

and

$$\theta_{G_k,s} = \begin{cases} \sqrt{2\ln(\#\boldsymbol{\rho}_{G_k,s}^{0})}\,\sigma_{G_k,s}, \sigma_{G_k,s} = \dfrac{\mathrm{median}(\mathrm{abs}(\boldsymbol{\epsilon}_{G_k,s}^{0}))}{0.6745}, \\ \quad \text{if } m_{j,k}^{i} \text{ belongs to a group } G_k \text{ of "local parameter"}; \\ \dfrac{c}{\sqrt{N_e}}, \text{ if } m_{j,k}^{i} \text{ belongs to a group } G_k \text{ of "global parameter"}. \end{cases} \tag{15}$$

In Eq. (13), f_{GC} represents the Gaspari–Cohn function (Gaspari and Cohn, 1999), as defined in Eq. (14); $\rho_{k,s}^{0}$ is the sample correlation between the kth model variable and the sth simulated observation data point, obtained from the ensembles of model variable $\left\{ m_{j,k}^{0} \right\}_{j=1}^{N_e}$ and simulated data point $\left\{ [\tilde{\mathbf{g}}\left(\mathbf{m}_j^0\right)]_s \right\}_{j=1}^{N_e}$, with $[\mathbf{x}]_s$ representing the sth element of a dummy vector $\mathbf{x}$[1]; $\theta_{G_k,s}$ corresponds to the threshold value with respect to a group G_k of model variables, to which the model variable $m_{j,k}^{i}$ belongs. In the localization scheme developed by Luo and Bhakta (2020), a model variable $m_{j,k}^{i}$ is classified as either a "local parameter" that is distributed on an associated numerical reservoir gridblock, or a "global parameter" that is not associated with any particular numerical gridblock. For instance, if $m_{j,k}^{i}$ corresponds to a porosity value, then it is considered as a "local parameter", and can be grouped together with porosity values on other numerical gridblocks to estimate a threshold value $\theta_{G_k,s}$. In this case, one applies the first line of Eq. (15), which combines the universal rule (Donoho and Johnstone, 1994) and the median absolute deviation (MAD) estimator (Donoho and Johnstone, 1995), to calculate a common threshold value $\theta_{G_k,s}$ for all model variables in the same group G_k. Here, $\boldsymbol{\epsilon}_{G_k,s}^{0}$ stands for the substitute sampling errors (which can be obtained by the random shuffle method proposed by Luo and Bhakta, 2020) in the correlation field $\boldsymbol{\rho}_{G_k,s}^{0} \equiv \left\{ \rho_{k,s}^{0} : k \in G_k \right\}$, and $\#\boldsymbol{\rho}_{G_k,s}^{0}$(equal to $\#\boldsymbol{\epsilon}_{G_k,s}^{0}$) means the number of elements of $\boldsymbol{\rho}_{G_k,s}^{0}$ or $\boldsymbol{\epsilon}_{G_k,s}^{0}$. Meanwhile, $m_{j,k}^{i}$ can also be a "global parameter", such as the end point value of a relative permeability curve. In this case, one can apply the result of the asymptotic distribution of a sample correlation to estimate $\theta_{G_k,s}$, thus leading to the second line of Eq. (15), where c is a positive coefficient whose recommended value can be located in, e.g., the interval [3, 4] (Luo and Xia, 2022) (also see the discussion in Luo and Bhakta, 2020). A side remark here is that in Eq. (15), the distinction between "local" and "global" parameters is neither strict nor even necessary. For instance, one can also apply the second line of Eq. (15) to estimate the threshold value $\theta_{G_k,s}$ for "local" parameters, resulting in only slightly worse history matching performance to the choice of using the first line of Eq. (15), as reported in Luo and Bhakta (2020). In the meantime, by applying the random shuffle method multiple times, one can also generate a number of substitute sampling errors for "global" parameters, making it possible to estimate the threshold value $\theta_{G_k,s}$ for a "global" parameter based on the first line of Eq. (15) (Ranazzi et al., 2022).

[1] Note that the iteration index i of the ensembles is set to 0, meaning that only the initial ensembles of reservoir models and corresponding simulated data points are used to calculate the tapering coefficient, for the reason explained in Luo et al. (2018).

2.3. Parameterized AutoAdaLoc scheme

In Eq. (15), both criteria to estimate threshold values for the AutoAdaLoc scheme are based on certain statistical analysis methods. Although our experience shows that the localization scheme established in this way appears to work reasonably well in a number of studies (Luo and Bhakta, 2020; Luo, 2021; Luo and Cruz, 2022; Ranazzi et al., 2022), it is possible to further improve its performance. Our main idea is that, instead of relying on a statistical analysis method to determine the threshold values, we parameterize the threshold values instead, and then use an ensemble-based CHOP procedure to optimize the parameterized localization scheme.

Concretely, in the parameterized AutoAdaLoc (P-AutoAdaLoc) scheme, we use the following formula to calculate the tapering coefficients:

$$t_{k,s} = f_{GC}\left(\frac{1 - \mathrm{abs}(\rho_{k,s}^{0})}{\ell_s} \right), \text{ for } k = 1, 2, \cdots, m; s = 1, 2, \cdots, p, \tag{16}$$

where $\ell_s > 0$ represents the localization length scale associated with the sth observation data point, and f_{GC} and $\rho_{k,s}^{0}$ remain the same as those in Eq. (13).

For Eq. (16), a more general parameterization strategy would be to replace the set of length scale parameters ℓ_s ($s = 1, 2, \cdots, p$) by a bigger one $\ell_{k,s}$ ($k = 1, 2, \cdots, m; s = 1, 2, \cdots, p$). However, the improved generality of this parameterization strategy is achieved at the expense of a much larger hyper-parameter size (equal to model size times data size), which may become computationally inefficient in practical reservoir data assimilation problems. In this regard, a lighter alternative is to let the localization length scale vary over different model variables instead, i.e., in the form of ℓ_k. While this alternative parameterization strategy appears feasible, it would still result in a larger set of hyper-parameters than the choice in Eq. (17), when the model size m is larger than the data size p, as in the numerical case studies later. Therefore, for the purpose of numerical efficiency, in the current work, we confine ourselves to the choice of optimizing ℓ_s ($s = 1, 2, \cdots, p$) (total number equal to data size).

Comparing Eqs. (13) and (16), the number of threshold values in Eq. (16) is seemingly less than that in Eq. (13). Nevertheless, an inspection on Eq. (15) tells that the threshold value computed by either the first or the second line would remain very close (or even identical). For instance, when using the first line to calculate the threshold value, the resulting threshold value is almost the same for different combinations of models variables and observation data points, with slight differences coming from statistical fluctuations in computing the standard deviation $\sigma_{G_k,s}$ through the MAD estimator (Donoho and Johnstone, 1995), as illustrated in Luo and Bhakta (2020). In contrast, the length scale parameters ℓ_s in Eq. (16) are independent of each other by design, which entails more degrees of freedom, and serves as the basis for further performance improvement in data assimilation, as will be illustrated later.

2.4. Continuous hyper-parameter optimization (CHOP) and its application to tune hyper-parameters in the P-AutoAdaLoc scheme

Equipping an IES with the P-AutoAdaLoc scheme means that the corresponding model update formula, Eq. (11) or (12), contains a set of localization length scales ℓ_s $(s = 1, 2, \cdots, p)$ that need to be specified. Note that these parameters do not come from a reservoir model, but originate in the P-AutoAdaLoc scheme instead. For this reason, we call these variables algorithmic hyper-parameters later.

Algorithmic hyper-parameters are often encountered in various situations. For instance, within the context of ensemble data assimilation, covariance inflation and localization (Anderson, 2009; Hamill et al., 2001) are two common techniques used to improve the performance of an EnKF, through which two hyper-parameters, namely, covariance inflation factor and localization length scale, are introduced into the EnKF algorithm. Meanwhile, in machine learning problems, there also exist various hyper-parameters, such as the learning rate and batch size, which need to be tuned in a stochastic optimization algorithm (Yu and Zhu, 2020).

In the current work, we follow a recently proposed CHOP procedure (Luo and Xia, 2022) to optimize the set of localization length scales ℓ_s. Unlike many existing hyper-parameter estimation methods in the literature (Yu and Zhu, 2020), the ensemble-based CHOP procedure aims to estimate multiple sets of hyper-parameters, rather than a single set. Under this setting, it can be shown that a CHOP problem can be converted into a conventional parameter estimation problem, and then be solved by an IES algorithm (Luo and Xia, 2022). As such, the CHOP procedure can be naturally integrated into an ensemble-based data assimilation workflow, while receiving certain practical benefits (e.g., being non-intrusive and derivative-free and providing a means of uncertainty quantification) similar to ensemble-based data assimilation methods.

Specifically, let $\mathbf{L}^i \equiv \left\{\boldsymbol{\ell}_j^i\right\}_{j=1}^{N_e}$ be an ensemble of N_e sets of localization length scales obtained at the ith iteration step, with $\boldsymbol{\ell}_j^i$ representing a single set of localization length scales used to update the jth reservoir model $\mathbf{m}_j^i$ to $\mathbf{m}_j^{i+1}$, as follows:

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \left(\mathbf{T}\left(\boldsymbol{\ell}_j^i\right) \circ \mathbf{K}^i\right)\left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}}\left(\mathbf{m}_j^i\right)\right), \tag{17}$$

where now the tapering matrix $\mathbf{T}\left(\boldsymbol{\ell}_j^i\right)$ is a function of $\boldsymbol{\ell}_j^i$, and thus depends on the indices of both iteration step i and ensemble member j. In this case, the element $t_{j,k,s}^i$ on the kth row and the sth column of the tapering matrix $\mathbf{T}\left(\boldsymbol{\ell}_j^i\right)$ is calculated by

$$t_{j,k,s}^i = f_{GC}\left(\frac{1 - \mathrm{abs}(\rho_{k,s}^0)}{\ell_{j,s}^i}\right), \quad \text{for } k = 1, 2, \cdots, m;\ s = 1, 2, \cdots, p;\ j = 1, 2, \cdots, N_e, \tag{18}$$

where $\ell_{j,s}^i$ $(s = 1, 2, \cdots, p)$ corresponds to the sth element of the set $\boldsymbol{\ell}_j^i$ of localization length scales.

Eqs. (17) and (18) constitute the model update formula, when the IES algorithm is equipped with the proposed P-AutoAdaLoc scheme. After this update step, we then proceed to update the ensemble $\mathbf{L}^i \equiv \left\{\boldsymbol{\ell}_j^i\right\}_{j=1}^{N_e}$ of localization length scales to a new ensemble $\mathbf{L}^{i+1} \equiv \left\{\boldsymbol{\ell}_j^{i+1}\right\}_{j=1}^{N_e}$. To this end, we exploit the notion in Luo and Xia (2022), which treats a generic model update formula as a mapping $\mathbf{f}$ that maps a prior model (also known as background) $\mathbf{m}^b$ to a posterior $\mathbf{m}^a$ (also known as analysis), for a given set of observations $\mathbf{d}^o$, a forward simulator $\mathbf{g}$ and a (possible) set of hyper-parameters $\boldsymbol{\theta}$, i.e.,

$$\begin{aligned}\mathbf{m}^a &= \mathbf{f}\left(\mathbf{m}^b, \boldsymbol{\theta} | \mathbf{d}^o, \mathbf{g}\right)\\ &\equiv \tilde{\mathbf{f}}\left(\mathbf{m}^b, \boldsymbol{\theta}\right).\end{aligned} \tag{19}$$

The first line of Eq. (19) is a mathematical abstraction/generalization of the model update formula in Eq. (17). Since in general the observations and forward simulator are not considered as variables, we drop these two quantities in the second line of Eq. (19) for notational convenience, and denote the corresponding mapping by $\tilde{\mathbf{f}}$ instead. With this abbreviation, we can re-express the IES update formula in Eq. (17) as

$$\mathbf{m}_j^{i+1} = \tilde{\mathbf{f}}\left(\mathbf{m}_j^i, \boldsymbol{\ell}_j^i\right). \tag{20}$$

A new ensemble $\mathbf{L}^{i+1} \equiv \left\{\boldsymbol{\ell}_j^{i+1}\right\}_{j=1}^{N_e}$ can then be obtained by solving a MAC problem similar to that in Eqs. (1) and (2), namely,

$$\min_{\{\boldsymbol{\ell}_j^{i+1}\}_{j=1}^{N_e}} \frac{1}{N_e}\sum_{j=1}^{N_e} \tilde{C}_j^{i+1}; \tag{21}$$

$$\begin{aligned}\tilde{C}_j^{i+1} \equiv &\left(\mathbf{d}_j^o - \mathbf{h}\left(\boldsymbol{\ell}_j^{i+1} | \mathbf{m}_j^i\right)\right)^T \mathbf{C}_d^{-1}\left(\mathbf{d}_j^o - \mathbf{h}\left(\boldsymbol{\ell}_j^{i+1} | \mathbf{m}_j^i\right)\right)\\ &+ \gamma^i\left(\boldsymbol{\ell}_j^{i+1} - \boldsymbol{\ell}_j^i\right)^T \left(\mathbf{C}_\ell^i\right)^{-1}\left(\boldsymbol{\ell}_j^{i+1} - \boldsymbol{\ell}_j^i\right);\end{aligned} \tag{22}$$

$$\mathbf{h}\left(\boldsymbol{\ell}_j^{i+1} | \mathbf{m}_j^i\right) \equiv \mathbf{g}\left(\tilde{\mathbf{f}}\left(\mathbf{m}_j^i, \boldsymbol{\ell}_j^{i+1}\right)\right), \tag{23}$$

where $\mathbf{C}_\ell^i$ in Eq. (22) is the sample covariance matrix of the ensemble $\mathbf{L}^i$ of localization length scales. In Eq. (23), $\mathbf{h}\left(\boldsymbol{\ell}_j^{i+1} | \mathbf{m}_j^i\right)$ can be considered as a function of $\boldsymbol{\ell}_j^{i+1}$ conditioned on the given reservoir model $\mathbf{m}_j^i$ (and $\mathbf{h}\left(\boldsymbol{\ell}_j^i | \mathbf{m}_j^i\right)$ in Eq. (24) later can be interpreted in a similar way). Following (Luo et al., 2015) (also cf. Eq. (3) – (5) of the current work), the MAC problem in Eq. (21) is approximately solved by the following hyper-parameter update formula:

$$\begin{aligned}\boldsymbol{\ell}_j^{i+1} &= \boldsymbol{\ell}_j^i + \mathbf{S}_\ell^i (\mathbf{S}_h^i)^T \left(\mathbf{S}_h^i (\mathbf{S}_h^i)^T + \gamma^i \mathbf{C}_d\right)^{-1}\left(\mathbf{d}_j^o - \mathbf{h}\left(\boldsymbol{\ell}_j^i | \mathbf{m}_j^i\right)\right)\\ &= \boldsymbol{\ell}_j^i + \mathbf{S}_\ell^i (\mathbf{S}_h^i)^T \left(\mathbf{S}_h^i (\mathbf{S}_h^i)^T + \gamma^i \mathbf{C}_d\right)^{-1}\left(\mathbf{d}_j^o - \mathbf{g}\left(\mathbf{m}_j^{i+1}\right)\right);\end{aligned} \tag{24}$$

$$\mathbf{S}_\ell^i = \frac{1}{\sqrt{N_e - 1}}\left[\boldsymbol{\ell}_1^i - \overline{\boldsymbol{\ell}}^i, \cdots, \boldsymbol{\ell}_{N_e}^i - \overline{\boldsymbol{\ell}}^i\right]; \quad \overline{\boldsymbol{\ell}}^i = \frac{1}{N_e}\sum_{j=1}^{N_e} \boldsymbol{\ell}_j^i; \tag{25}$$

$$\begin{aligned}\mathbf{S}_h^i &= \frac{1}{\sqrt{N_e - 1}}\left[\mathbf{h}(\boldsymbol{\ell}_1^i | \mathbf{m}_1^i) - \mathbf{h}(\overline{\boldsymbol{\ell}}^i | \overline{\mathbf{m}}^i), \cdots, \mathbf{h}(\boldsymbol{\ell}_{N_e}^i | \mathbf{m}_{N_e}^i) - \mathbf{h}(\overline{\boldsymbol{\ell}}^i | \overline{\mathbf{m}}^i)\right];\\ &\approx \frac{1}{\sqrt{N_e - 1}}\left[\mathbf{g}(\mathbf{m}_1^{i+1}) - \mathbf{g}(\overline{\mathbf{m}}^{i+1}), \cdots, \mathbf{g}(\mathbf{m}_{N_e}^{i+1}) - \mathbf{g}(\overline{\mathbf{m}}^{i+1})\right]\\ &= \mathbf{S}_g^{i+1}.\end{aligned} \tag{26}$$

In Eqs. (24) and (26), the second lines are derived based on Eqs. (20) and (23), with $\mathbf{h}\left(\boldsymbol{\ell}_j^i | \mathbf{m}_j^i\right) = \mathbf{g}\left(\tilde{\mathbf{f}}\left(\mathbf{m}_j^i, \boldsymbol{\ell}_j^i\right)\right) = \mathbf{g}\left(\mathbf{m}_j^{i+1}\right)$ for $j = 1, 2, \cdots, N_e$. In addition, in Eq. (26) the approximation $\mathbf{h}\left(\overline{\boldsymbol{\ell}}^i | \overline{\mathbf{m}}^i\right) = \mathbf{g}\left(\tilde{\mathbf{f}}\left(\overline{\mathbf{m}}^i, \overline{\boldsymbol{\ell}}^i\right)\right) \approx \mathbf{g}(\overline{\mathbf{m}}^{i+1})$ is adopted, mainly for the purpose to avoid running an extra forward reservoir simulation for the reservoir model produced by the update $\tilde{\mathbf{f}}\left(\overline{\mathbf{m}}^i, \overline{\boldsymbol{\ell}}^i\right)$.

Inserting Eq. (26) into Eq. (24), one has

$$\boldsymbol{\ell}_j^{i+1} \approx \boldsymbol{\ell}_j^i + \mathbf{S}_\ell^i (\mathbf{S}_g^{i+1})^T \left(\mathbf{S}_g^{i+1} (\mathbf{S}_g^{i+1})^T + \gamma^i \mathbf{C}_d\right)^{-1}\left(\mathbf{d}_j^o - \mathbf{g}\left(\mathbf{m}_j^{i+1}\right)\right). \tag{27}$$

Following Eq. (6) – (9), normalization in the observation space and TSVD can also be introduced to Eq. (27), although we skip the details here for brevity. Without loss of generality, one can also re-express Eq. (27) as something like (cf Eq. (10)):

$$\boldsymbol{\ell}_j^{i+1} = \boldsymbol{\ell}_j^i + \mathbf{K}_\ell^i\left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}}\left(\mathbf{m}_j^{i+1}\right)\right), \tag{28}$$

where $\mathbf{K}_\ell^i$ is a certain Kalman-gain like matrix, e.g.,

$$\mathbf{K}_\ell^i = \mathbf{S}_\ell^i (\tilde{\mathbf{S}}_g^{i+1})^T \left(\tilde{\mathbf{S}}_g^{i+1} (\tilde{\mathbf{S}}_g^{i+1})^T + \gamma^i \mathbf{I}_p\right)^{-1}, \tag{29}$$

when normalization in the observation space is conducted.

Like an ordinary IES algorithm, Eq. (28) also needs localization to tackle the adversarial effects of a small ensemble size. To this end, we also replace the Kalman-gain like matrix $\mathbf{K}_\ell^i$ by the Schur product between $\mathbf{K}_\ell^i$ and a tapering matrix $\mathbf{T}_\ell\left(\boldsymbol{\ell}_j^i\right)$, which is constructed in exactly the same way (cf. Eq. (18)) as $\mathbf{T}\left(\boldsymbol{\ell}_j^i\right)$ in Eq. (10), but has

Algorithm 1 Pseudo-code of the IES algorithm equipped with the original AutoAdaLoc localization scheme. The texts in red highlight places where differences take place, in comparison to the pseudo-code in Algorithm 2.

Require: Initial ensembles of reservoir models $\mathbf{M}^0 = \{\mathbf{m}_j^0\}_{j=1}^{N_e}$, simulated observations $\{\mathbf{g}(\mathbf{m}_j^0)\}_{j=1}^{N_e}$ and $\mathbf{g}(\overline{\mathbf{m}}^0)$; Initial regularization parameter γ^0

Construct the tapering matrix $\mathbf{T}$ ▷ cf. Eqs. (13)–(15)

Iteration index $i \leftarrow 0$; IES not stopped $\leftarrow$ True

while IES not stopped **do**

  Calculate the Kalman-gain like matrix $\mathbf{K}^i$ for reservoir model update, e.g.,

$$\mathbf{K}^i = \mathbf{S}_m^i (\tilde{\mathbf{S}}_g^i)^T \left(\tilde{\mathbf{S}}_g^i (\tilde{\mathbf{S}}_g^i)^T + \gamma^i \mathbf{I}_p \right)^{-1} \quad \triangleright \text{ cf. Eqs. (4)–(9)}$$

  Update the reservoir models $\mathbf{m}_j^i$, e.g.,

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \left(\mathbf{T} \circ \mathbf{K}^i\right) \left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}} \left(\mathbf{m}_j^i \right) \right) \quad \triangleright \text{ cf. Eq. (11)}$$

  Run reservoir simulations to get the simulated observations $\{\mathbf{g}(\mathbf{m}_j^{i+1})\}_{j=1}^{N_e}$ and $\mathbf{g}(\overline{\mathbf{m}}^{i+1})$,

  Check data mismatch and update the value of γ^i ▷ cf. Luo et al. (2015)

  Apply stopping criteria to decide whether to stop the IES or not ▷ cf. Luo et al. (2015)

  $i \leftarrow i + 1$

end while

a different number of matrix rows. As a result, we end up with the following "tapered" hyper-parameter update formula:

$$\boldsymbol{\ell}_j^{i+1} = \boldsymbol{\ell}_j^i + \left(\mathbf{T}_\ell \left(\boldsymbol{\ell}_j^i \right) \circ \mathbf{K}_\ell^i \right) \left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}} \left(\mathbf{m}_j^{i+1} \right) \right). \tag{30}$$

Eqs. (17) and (30) thus constitute the main formulae that apply an IES algorithm to update both reservoir model variables and algorithmic hyper-parameters in an alternative way. Compared to the localized model update formula Eq. (11) in the original IES algorithm, a noticeable feature in Eqs. (17) and (30) is that the tapering matrices $\mathbf{T}\left(\boldsymbol{\ell}_j^i\right)$ and $\mathbf{T}_\ell\left(\boldsymbol{\ell}_j^i\right)$ therein depend on individual members $\boldsymbol{\ell}_j^i$ of the ensemble of localization length scales. This means that the model/hyper-parameter update formula in Eq. (17)/Eq. (30) also depends on individual ensemble members. While this feature can lead to improved data assimilation performance as will be illustrated below, in general, it also entails increased computational cost. Later we will discuss a few possible strategies to help reduce the overall computational cost.

3. Case studies

3.1. Case 1: Five spots (5Spots)

We start with a relatively small 2D case study, in which a reservoir model contains an inverted five-spots (5Spots) well pattern with four producers on the corners and one injector in the centre, as indicated in Fig. 1. In terms of the number of gridblocks, the dimension of the reservoir model is 50 × 50. The four producers are located at the coordinates (2,2) (upper left, labelled as P1 hereafter), (49, 2) (upper right, labelled as P2 hereafter), (2, 49) (lower left, labelled as P3 hereafter), (49, 49) (lower right, labelled as P4 hereafter), respectively; whereas the injector sits at the centre with the coordinate (25, 25) (labelled as I1 hereafter). In numerical simulations, the producers P1–P3 and the injector I1 are set to be open, but the producer P4 is shut off.

In this case study, the unknown parameters are porosity (PORO) and x-directional permeability (PERMX) in the scale of the natural logarithm. The phases present in the reservoir include oil, water, and gas. The PERMX and PORO maps of the reference model are shown in Fig. 1(a) and Fig. 2(a), respectively. The reference model is used to generate a set of production data every 30 days, for a total period of 1500 days with 50 report steps. At each reporting time, the types of production data include oil, water, and gas production rates (denoted by OPR, WPR, and GPR, respectively) from P1–P3, and bottom hole pressure (BHP) from P1–P3 and I1. As such, there are 13 data points at each reporting time, and 650 data points (13 × 50) during the whole time window. Subsequently, some zero-mean Gaussian white noise is generated to mimic the potential presence of observation errors in practical data assimilation problems, and then added to the production data from the reference model. Specifically, the standard deviations (STDs) of the observation errors for OPR, WPR, and GPR data are set to be 10% of the magnitudes of respective data,[2] whereas the STDs of the observation errors for BHP data are set to be 1 Bar.

An initial ensemble of 100 reservoir models is generated through sequential Gaussian simulation. The mean PERMX and PORO maps of the initial ensemble are shown in Figs. 1(b) and 2(b), respectively. The IES algorithm of Luo et al. (2015) is adopted in all numerical examples hereafter to iteratively update the ensembles of reservoir models. The maximum iteration step is set to be 20, while the IES will stop if the maximum iteration step is reached, or if the relative change of the average data mismatch (see Eq. (32) later for its definition) at two consecutive iteration steps is less than 0.01%.

In this case study we investigate and compare the performance of four different choices of localization schemes, which correspond to the following settings:

Setting (1): No localization is adopted in the IES algorithm (labelled as "noLoc" hereafter);

Setting (2): The AutoAdaLoc scheme based on Eq. (12)–(15) is adopted. Note that in this case the threshold values $\theta_{G_k,s}$ are preset and fixed during the iteration steps;

Setting (3): The P-AutoAdaLoc scheme based on Eq. (17)–(18) is adopted. In this case, for a given iteration step index i and ensemble member index j, the set of localization length scales $\boldsymbol{\ell}_j^i$ only contains one element ℓ_j^i, whose value is shared among all observation elements. In other words, in Eq. (18) one has the constraint $\ell_{j,1}^i = \ell_{j,2}^i = \cdots = \ell_{j,650}^i$ (with the number of observations $p = 650$), and $\ell_{j,s}^i = \ell_j^i$, $\forall s = 1, 2, \cdots, 650$. Note that ℓ_j^i will be updated through the CHOP procedure (cf. Eq. (30)), and they may thus vary over the iteration steps and ensemble members in general. For reference later, this localization scheme is labelled as "P-AutoAdaLoc-1".

Setting (4): The setting here is the same as Setting (3), except that the set of localization length scales $\boldsymbol{\ell}_j^i$ contains p elements $\ell_{j,s}^i$, $s = 1, 2, \cdots, 650$, which would be distinct from each other in general. For distinction later, this localization scheme is labelled as "P-AutoAdaLoc-0".

[2] If the magnitude of a data point is zero, then the corresponding STD value is set to 10^{-6} instead to avoid the numerical issue of division by zero in data assimilation later.

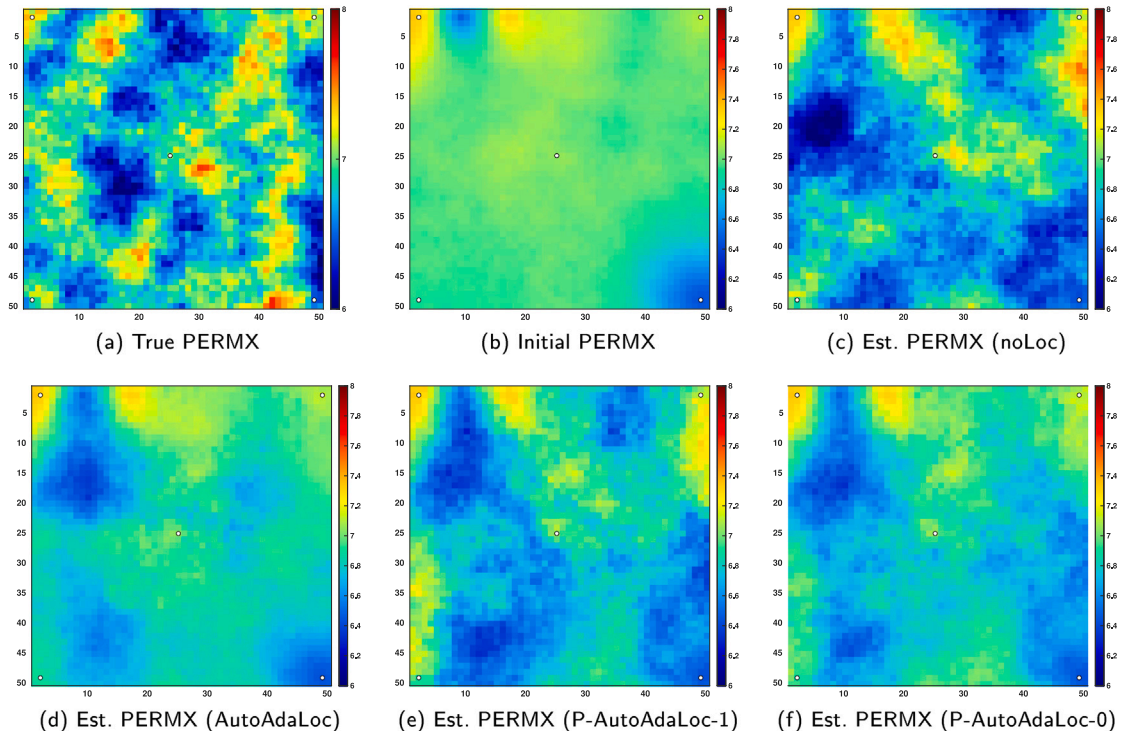


Fig. 1. PERMX maps with respect to (a) the reference model; (b) the mean of the initial ensemble; (c) – (f) the means of the final estimated (est.) ensembles obtained by the IES algorithm with no localization scheme (labelled as "noLoc"), the AutoAdaLoc scheme, the P-AutoAdaLoc scheme with a single common length scale for all observation elements (labelled as "P-AutoAdaLoc-1"), and the P-AutoAdaLoc scheme with an individual length scale for each observation element (labelled as "P-AutoAdaLoc-0"), respectively, in the 5Spots case. In all the maps, the small white circles indicate the locations of injection (in the centre) or production (on the corners) wells.

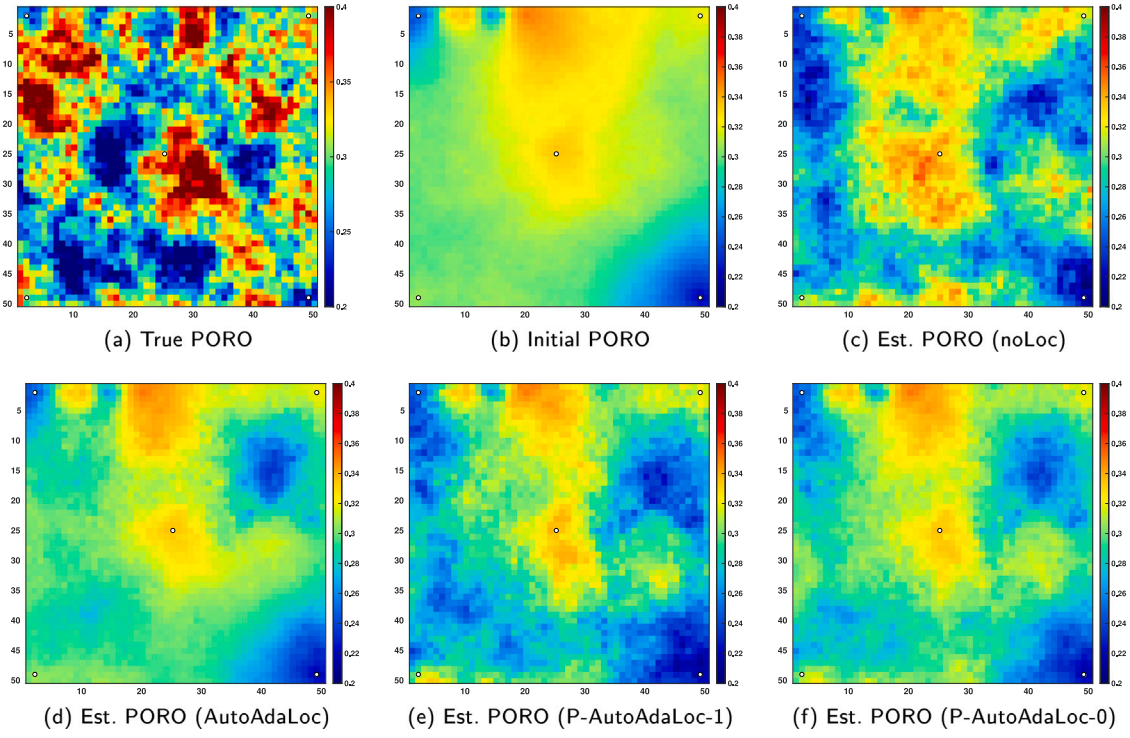


Fig. 2. As in Fig. 1, but for PORO maps in the 5Spots case.

Algorithm 2 Pseudo-code of the IES algorithm equipped with the P-AutoAdaLoc localization scheme. The texts in red highlight places where differences take place, in comparison to the pseudo-code in Algorithm 1.

Require: Initial ensembles of reservoir models $\mathbf{M}^0 = \{\mathbf{m}_j^0\}_{j=1}^{N_e}$, simulated observations $\{\mathbf{g}(\mathbf{m}_j^0)\}_{j=1}^{N_e}$ and $\mathbf{g}(\overline{\mathbf{m}}^0)$, localization length scales $\mathbf{L}^0 = \left\{\boldsymbol{\ell}_j^0\right\}_{j=1}^{N_e}$; Initial regularization parameter γ^0

Iteration index $i \leftarrow 0$; IES not stopped $\leftarrow$ True
while IES not stopped **do**
  Calculate the Kalman-gain like matrix $\mathbf{K}^i$ for reservoir model update, e.g.,
    $\mathbf{K}^i = \mathbf{S}_m^i(\tilde{\mathbf{S}}_g^i)^T\left(\tilde{\mathbf{S}}_g^i(\tilde{\mathbf{S}}_g^i)^T + \gamma^i\mathbf{I}_p\right)^{-1}$ ▷ cf. Eqs. (4)–(9)
  for $j = 1, 2, \cdots, N_e$ **do**
    Construct the tapering matrix $\mathbf{T}\left(\boldsymbol{\ell}_j^i\right)$ with the set of length scales $\boldsymbol{\ell}_j^i$ ▷ cf. Eq. (18)
    Update the reservoir models $\mathbf{m}_j^i$, e.g.,
      $\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \left(\mathbf{T}\left(\boldsymbol{\ell}_j^i\right) \circ \mathbf{K}^i\right)\left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}}\left(\mathbf{m}_j^i\right)\right)$ ▷ cf. Eq. (17)
  end for
  Run reservoir simulations to get the simulated observations $\{\mathbf{g}(\mathbf{m}_j^{i+1})\}_{j=1}^{N_e}$ and $\mathbf{g}(\overline{\mathbf{m}}^{i+1})$,
  Calculate the Kalman-gain like matrix $\mathbf{K}_\ell^i$ for hyper-parameter update, e.g.,
    $\mathbf{K}_\ell^i = \mathbf{S}_\ell^i(\tilde{\mathbf{S}}_g^{i+1})^T\left(\tilde{\mathbf{S}}_g^{i+1}(\tilde{\mathbf{S}}_g^{i+1})^T + \gamma^i\mathbf{I}_p\right)^{-1}$ ▷ cf. Eqs. (24) – (29)
  for $j = 1, 2, \cdots, N_e$ **do**
    Construct the tapering matrix $\mathbf{T}_\ell\left(\boldsymbol{\ell}_j^i\right)$ with the set of length scales $\boldsymbol{\ell}_j^i$ ▷ similar to Eq. (18)
    Update the set of localization length scale $\boldsymbol{\ell}_j^i$, e.g.,
      $\boldsymbol{\ell}_j^{i+1} = \boldsymbol{\ell}_j^i + \left(\mathbf{T}_\ell\left(\boldsymbol{\ell}_j^i\right) \circ \mathbf{K}_\ell^i\right)\left(\tilde{\mathbf{d}}_j^o - \tilde{\mathbf{g}}\left(\mathbf{m}_j^{i+1}\right)\right)$ ▷ cf. Eq. (30)
  end for
  Check data mismatch and update the value of γ^i ▷ cf. Luo et al. (2015)
  Apply stopping criteria to decide whether to stop the IES or not ▷ cf. Luo et al. (2015)
  $i \leftarrow i + 1$
end while

Table 1
Data mismatch (DM), root mean squared error (RMSE), and ensemble spread obtained by the IES algorithm in the 5Spots case study, when it is equipped with the noLoc, AutoAdaLoc, P-AutoAdaLoc-1 or P-AutoAdaLoc-0 scheme. Note that the mean and standard deviation (STD) values of DM or RMSE are computed with respect to the members of either the initial or the final ensemble, and that the lowest mean DM, mean RMSE and spread values are highlighted by the bold font.

	Data mismatch (mean ± STD)	RMSE of (log) PERMX (mean ± STD)	RMSE of PORO (mean ± STD)	Total RMSE (mean ± STD)	Spread
Initial ensemble	$(1.3706 \pm 4.0594) \times 10^{10}$	0.4367 ± 0.0263	0.0644 ± 0.0037	0.3121 ± 0.0184	0.1413
Final ensemble (noLoc)	**663.7167 ± 34.2293**	0.4332 ± 0.0230	0.0681 ± 0.0031	0.3101 ± 0.0162	**0.0862**
Final ensemble (AutoAdaLoc)	$(3.3649 \pm 1.5646) \times 10^3$	0.4009 ± 0.0228	**0.0640 ± 0.0033**	0.2871 ± 0.0160	0.1167
Final ensemble (P-AutoAdaLoc-1)	$(1.3424 \pm 0.6089) \times 10^3$	0.3961 ± 0.0235	0.0656 ± 0.0036	0.2839 ± 0.0165	0.0989
Final ensemble (P-AutoAdaLoc-0)	$(2.2806 \pm 0.6770) \times 10^3$	**0.3920 ± 0.0265**	0.0651 ± 0.0034	**0.2810 ± 0.0186**	0.1109

Although the focus of the current work is to compare the performance of the AutoAdaLoc scheme in Setting (2) and the P-AutoAdaLoc-0 scheme in Setting (4), we choose to additionally consider the P-AutoAdaLoc-1 scheme in the current case study (while using the noLoc setting as the baseline). The reason behind this choice is that essentially Settings (2) and (3) can be considered as similar localization schemes, since the threshold values $\theta_{G_k,s}$ in the AutoAdaLoc scheme (cf. Eq. (15)) are almost the same for different combinations of the indices G_k and s. From this perspective, the role of the quantity $1 - \theta_{G_k,s}$ in the AutoAdaLoc scheme (cf. Eq. (13)) is similar to that of the length scale ℓ_j^i in the P-AutoAdaLoc-1 scheme (cf. Eq. (18), with $\ell_{j,s}^i = \ell_j^i$, $\forall s = 1, 2, \cdots, p$). An important difference is that the values of $1 - \theta_{G_k,s}$ in the AutoAdaLoc scheme are pre-determined based on certain statistical analysis, whereas ℓ_j^i in the P-AutoAdaLoc-1 scheme are considered as hyper-parameters and thus optimized through the CHOP procedure. As such, a comparison between the AutoAdaLoc and P-AutoAdaLoc-1 schemes would be useful for reflecting the impacts of the CHOP procedure on optimizing a single localization length scale.

Meanwhile, since we adopt similar abbreviations "P-AutoAdaLoc", "P-AutoAdaLoc-0" and "P-AutoAdaLoc-1" in various contexts of the current work, it appears useful for us to explain the typical scenarios in which these abbreviations are used. Roughly speaking, "P-AutoAdaLoc" refers to a generic parameterized AutoAdaLoc scheme. Following the discussion in Section 2.3, there could be diverse parameterization strategies that can result in different numbers of hyper-parameters in parameterized AutoAdaLoc schemes, whereas "P-AutoAdaLoc-0" and "P-AutoAdaLoc-1" correspond to two particular parameterization strategies, as explained in Settings (3) and (4). In other words, one can consider "P-AutoAdaLoc-0" and "P-AutoAdaLoc-1" as two concrete implementations of the "P-AutoAdaLoc" scheme.

In line with Algorithm 2, in Setting (3) or (4), an initial ensemble of localization length scale(s) $\mathbf{L}^0 = \left\{\boldsymbol{\ell}_j^0\right\}_{j=1}^{N_e}$ is generated as follows: For each ensemble member $\boldsymbol{\ell}_j^0$, each of its component $\ell_{j,s}^0$ ($s = 1$ in the P-AutoAdaLoc-1 scheme, and $s = 1, 2, \cdots, 650$ in the P-AutoAdaLoc-0 scheme) is drawn as an independent and identically distributed (i.i.d) sample from the uniform distribution on the interval $[0.23, 0.43]$. Note that here the aforementioned interval is chosen manually. In principle, the upper and lower boundaries of the interval can also be considered as some hyper-parameters and be optimized through a certain CHOP procedure, which, however, is not considered in the current work. For illustration, the first column of Fig. 3 shows the histograms of some localization length scales from the initial ensembles in both the P-AutoAdaLoc-1 and the P-AutoAdaLoc-0 schemes.

We evaluate three quantities, namely, data mismatch (DM), root mean squared error (RMSE) and ensemble spread (spread for short),

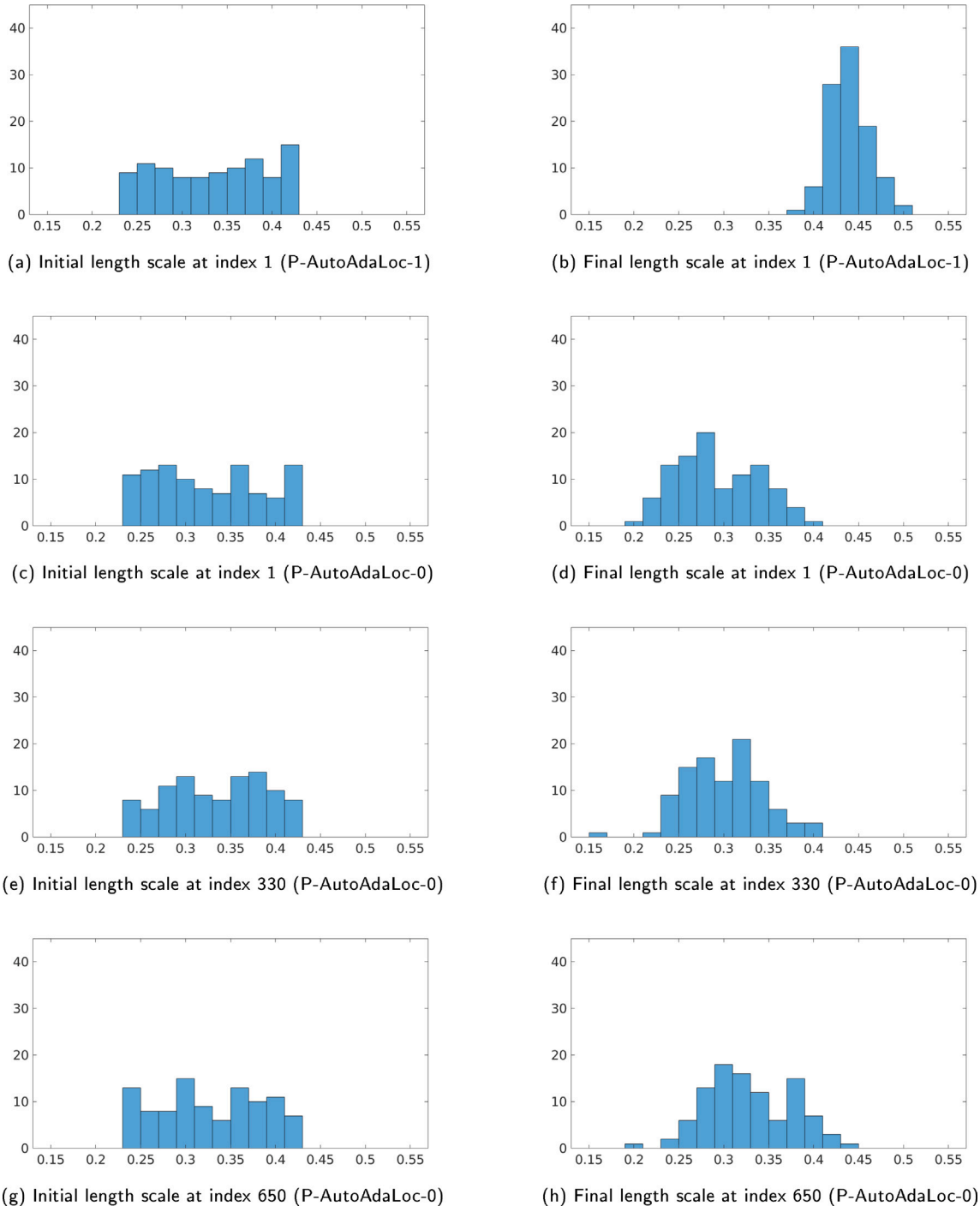


Fig. 3. Histograms of the initial (first column) and final (second column) ensembles of localization length scales in the P-AutoAdaLoc-1 and P-AutoAdaLoc-0 schemes used in the 5Spots case. Histograms on the first row depict the distributions of the single localization length scale in the P-AutoAdaLoc-1 scheme, whereas those on the second to the fourth rows show the distributions of the 1st, the 330th, and the 650th localization length scale, respectively, in the P-AutoAdaLoc-0 scheme.

in the synthetic case study. DM (RMSE) is adopted to quantify the distance between simulated observations (estimated reservoir model) and real observations (reference reservoir model), whereas spread is used to measure ensemble variability.

For the definition of RMSE, suppose that $\mathbf{m}$ is an m-dimensional reservoir model, and $\mathbf{m}^{ref}$ is the reference reservoir model (ground truth) that generates the noisy observation $\mathbf{d}^o$. Then the RMSE of $\mathbf{m}$, with respect to the reference model $\mathbf{m}^{ref}$, is given by

$$\mathrm{RMSE}\left(\mathbf{m};\mathbf{m}^{ref}\right) \equiv \|\mathbf{m}-\mathbf{m}^{ref}\|_2/\sqrt{m}, \tag{31}$$

where the operator $\|\bullet\|_2$ returns the euclidean norm of its operand $\bullet$.

Meanwhile, the DM of $\mathbf{m}$, with respect to the observation $\mathbf{d}^o$, is defined as

$$\mathrm{DM}\left(\mathbf{m};\mathbf{d}^o\right) \equiv \left(\mathbf{d}^o-\mathbf{g}\left(\mathbf{m}\right)\right)^T \mathbf{C}_d^{-1}\left(\mathbf{d}^o-\mathbf{g}\left(\mathbf{m}\right)\right). \tag{32}$$

Regarding the definition of ensemble spread, let $\mathcal{M} = \left\{\mathbf{m}_j \equiv \left[m_{j,1}, m_{j,2}, \cdots m_{j,m}\right]^T\right\}_{j=1}^{N_e}$ be an ensemble of estimated reservoir models, with $m_{j,k}$ denoting the kth element of $\mathbf{m}_j$ $(k = 1, 2, \cdots, m)$. In this case, one can first construct a vector $\mathbf{S} \equiv \left[\sigma_1, \sigma_2, \cdots, \sigma_m\right]^T$, with σ_k being

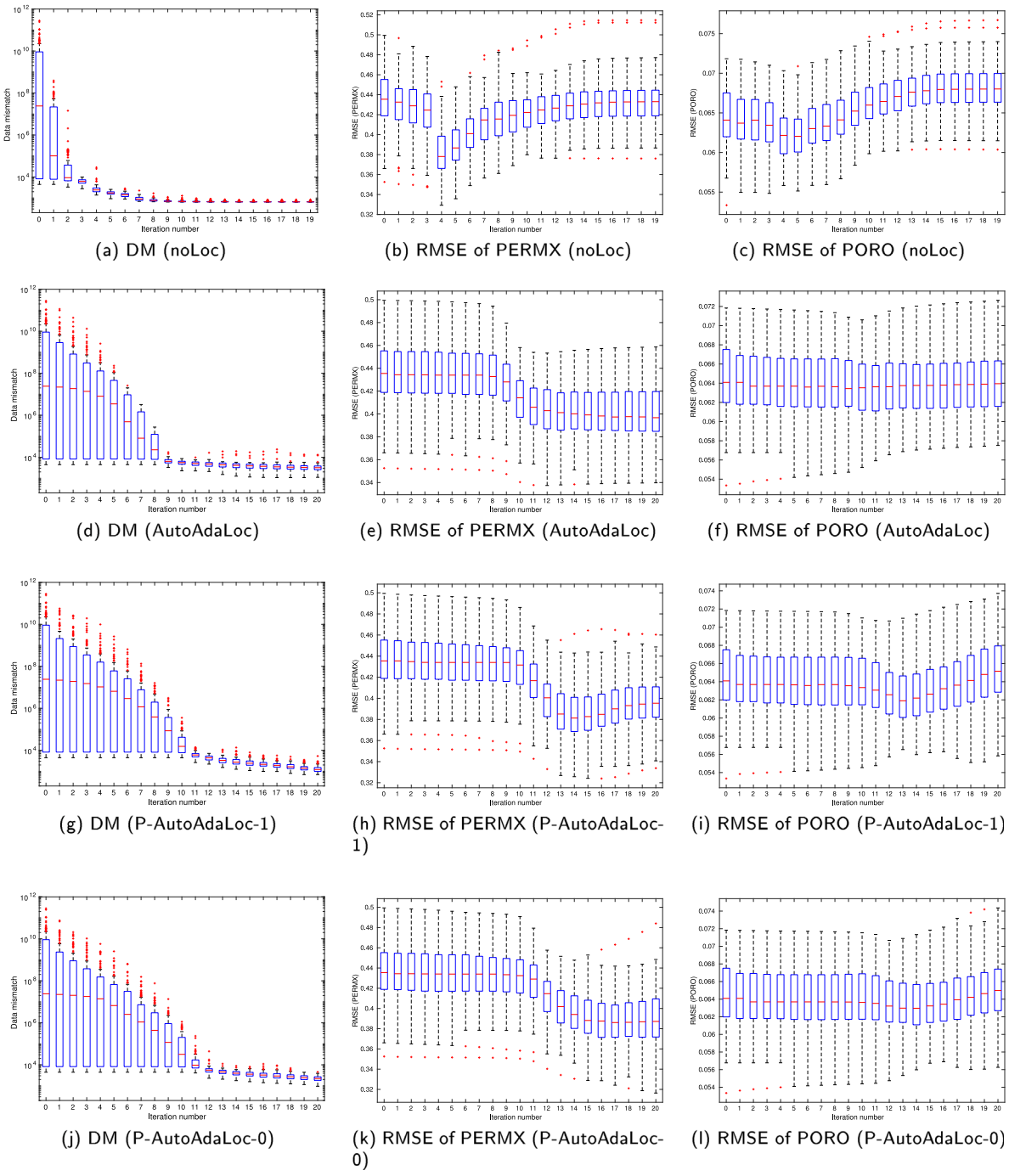


Fig. 4. Box plots of DM (first column) and RMSE of PERMX (second column) and PORO (third column) at different iteration steps of the IES algorithm, which is equipped with noLoc (first row), AutoAdaLoc (second row), P-AutoAdaLoc-1 (third row) and P-AutoAdaLoc-0 (fourth row) scheme, respectively, in the 5Spots case.

the sample standard deviation of the ensemble $\{m_{j,k}\}_{j=1}^{N_e}$. Then, the ensemble spread is computed as

$$S_{en}(\mathcal{M}) = \|\mathbf{S}\|_2/\sqrt{m}. \tag{33}$$

With the above definitions, Table 1 reports the DM and RMSE values (in the form of mean ± STD), as well as ensemble spreads, with respect to the initial ensemble, and the final ensembles obtained by the IES algorithm, when it is equipped with noLoc, AutoAdaLoc, P-AutoAdaLoc-1 or P-AutoAdaLoc-0 scheme. In comparison to the initial ensemble, the final ensembles obtained by the IES algorithm correspond to several-order lower mean DM values. Among the four choices of localization schemes, noLoc leads to the lowest mean DM. Nevertheless, in terms of mean total RMSE (i.e., the mean RMSE of log PERMX and PORO together), noLoc results in the highest mean RMSE, which is around only 0.6% lower than the mean RMSE of the initial ensemble. This suggests that the noLoc scheme tends to overfit the observations, which in turn may lead to a deteriorated performance in terms of RMSE, as was also observed in other work (Luo et al., 2017). In contrast, although the mean DM values of the other three localization schemes are higher, the mean values of the corresponding total RMSEs become lower than that of noLoc instead.

Moreover, in terms of ensemble spread, it can be seen from Table 1 that the final ensembles of all four localization schemes maintain sufficiently large values, meaning that there are substantial varieties in the final ensembles, and that ensemble collapse does appear to be an

issue in this case study. Among the four localization schemes, noLoc leads to the lowest spread value (as expected), AutoAdaLoc achieves the highest spread, and P-AutoAdaLoc-1 and P-AutoAdaLoc-0 obtain intermediate results.

Meanwhile, an inspection of the RMSE values of PERMX and PORO in Table 1 indicates that in this particular case study, the P-AutoAdaLoc-1 and P-AutoAdaLoc-0 schemes have somewhat different characteristics than the AutoAdaLoc scheme: For the latter, its mean RMSE values of PERMX and PORO are both lower than the corresponding values of the initial ensemble. For the P-AutoAdaLoc-1 and P-AutoAdaLoc-0 schemes, their mean RMSE values of PERMX are lower than the corresponding values of the initial ensemble and the final ensemble of the AutoAdaLoc scheme, but their mean RMSE values of PORO are slightly higher than those of the initial ensemble and the final ensemble of the AutoAdaLoc scheme instead. A possible reason behind this different behaviour of the P-AutoAdaLoc schemes is that in this particular case study, there is stronger causality relation between PERMX and production data, but weaker one between PORO and production data, as was similarly observed in Luo et al. (2017). When applying the CHOP procedure to the P-AutoAdaLoc schemes, the stronger causality relation has more dominant impacts on the update of localization length scales. As a result, the P-AutoAdaLoc schemes result in lower mean RMSE values for PERMX, but higher mean RMSE values for PORO. Putting together PERMX and PORO, one can see that overall the P-AutoAdaLoc scheme with either single (P-AutoAdaLoc-1) or multiple (P-AutoAdaLoc-0) localization length scale(s) leads to lower mean total RMSE than that of the AutoAdaLoc scheme, whereas the P-AutoAdaLoc-0 scheme with multiple localization length scales performs better than the P-AutoAdaLoc-1 scheme with a single localization length scale. This suggests that in this particular case study, it appears beneficial to introduce tunable length scales to a localization scheme, and that an increased number of tunable length scales tends to result in further reduced RMSE values.

Complementary to the results in Table 1, Fig. 4 shows box plots of DM and RMSE of PERMX and PORO at different iteration steps of the IES algorithm, in which the noLoc, AutoAdaLoc, P-AutoAdaLoc-1 or P-AutoAdaLoc-0 scheme is adopted. With the noLoc scheme, DM tends to reduce at a faster pace and to a lower amount. In terms of RMSE of PERMX and PORO, however, there is a clear V-shaped behaviour in the sense that at the first few iteration steps, the RMSE values tend to decrease; after that, the RMSE values uprise and become close to or higher than the corresponding values of the initial ensemble, indicating the possibility of over-fitting the observations. For the AutoAdaLoc, P-AutoAdaLoc-1, and P-AutoAdaLoc-0 schemes, their corresponding box plots of DM exhibit somewhat similar behaviour: the DM values change slower, and stay higher at the end of the iteration process, than those with the noLoc scheme. In terms of RMSE, the values of both PERMX and PORO with respect to the AutoAdaLoc, P-AutoAdaLoc-1, and P-AutoAdaLoc-0 schemes appear lower than those with respect to the noLoc scheme. The P-AutoAdaLoc-0 scheme leads to the lowest RMSE of PERMX, whereas the AutoAdaLoc scheme achieves the lowest RMSE of PORO. In terms of total RMSE, the P-AutoAdaLoc-1 and P-AutoAdaLoc-0 schemes perform better than the AutoAdaLoc and noLoc schemes. For the P-AutoAdaLoc-1 scheme, the box plots exhibit V-shaped behaviour for RMSE of both PERMX and PORO. This phenomenon is mitigated with the P-AutoAdaLoc-0 scheme, suggesting that in this particular case study, it appears more beneficial to include multiple localization length scales than to use a single localization length scale in the P-AutoAdaLoc scheme.

Additionally, Figs. 1 and 2 disclose the PERMX and PORO maps with respect to the reference model, and the mean models of the initial and final ensembles. Given the limited spatial coverage of the production data, the IES algorithm with either localization scheme is not able to obtain estimations of PERMX and PORO maps that resemble the reference maps to a satisfactory extent, although the mean maps of the final ensembles do appear better than those of the initial ensemble. Among the four localization schemes, it appears that the final mean maps obtained by the AutoAdaLoc, P-AutoAdaLoc-1, and P-AutoAdaLoc-0 schemes are more similar to each other due to the similarities among the adopted localization schemes, whereas those obtained by the noLoc scheme have more distinctions because of the absence of localization.

Furthermore, Fig. 3 indicates the histograms with respect to the initial and final ensembles of hyper-parameters in the P-AutoAdaLoc-1 and P-AutoAdaLoc-0 schemes. For the P-AutoAdaLoc-1 scheme, each ensemble member $\boldsymbol{\ell}_j^i$ ($i = 0$ or $i = 20$) only has a single element $\ell_{j,1}^i$, whereas for the P-AutoAdaLoc-0 scheme used in this particular case, each ensemble member $\boldsymbol{\ell}_j^i$ contains 650 elements $\ell_{j,s}^i$ ($s = 1, 2, \cdots, 650$) instead. For brevity, Fig. 3 only reports the histograms of the ensembles of the 1st ($\{\ell_{j,1}^i\}_{j=1}^{100}$), the 330-th ($\{\ell_{j,330}^i\}_{j=1}^{100}$) and the 650-th elements ($\{\ell_{j,650}^i\}_{j=1}^{100}$), respectively. Comparing the histograms of the P-AutoAdaLoc-0 and P-AutoAdaLoc-1 schemes, it can be seen that there are noticeable differences (in terms of locations and heights) among the histograms of the initial (first column) and final (second column) ensembles. However, the differences in the P-AutoAdaLoc-0 scheme appear to be less substantial than those in the P-AutoAdaLoc-1 scheme. A possible reason behind the more significant changes of the hyper-parameters in the P-AutoAdaLoc-1 scheme is that it only contains a single localization length scale in each ensemble member, which implies less degree of freedom for the purpose of fitting observations. As such, it has to make more significant adjustments of localization length scales in order to lower DM values during data assimilation.

Putting together the results in Table 1 and Figs. 1–4, we conclude that in this particular case study, the AutoAdaLoc, P-AutoAdaLoc-1, and P-AutoAdaLoc-0 schemes surpass the noLoc scheme. The performance of the AutoAdaLoc, P-AutoAdaLoc-1, and P-AutoAdaLoc-0 schemes appears close to each other. In terms of the total RMSE, the P-AutoAdaLoc-0 scheme performs better than the P-AutoAdaLoc-1 scheme, whereas the latter appears slightly better than the AutoAdaLoc scheme. Taking into account these results and for brevity later, in subsequent investigations, we will be focused on further comparing the performance of the AutoAdaLoc and P-AutoAdaLoc-0 schemes in two more case studies.

3.2. Case 2:Multiple five spots (M-5Spots)

The second example considers a larger 2D case study investigated by Chen and Oliver (2010), in which the reservoir model contains multiple inverted 5-spots well (M-5Spots) patterns, with each injector surrounded by four producers. In terms of reservoir gridblocks, the dimension of the reservoir model is 167×167. For the injectors, both their horizontal and vertical coordinates are integers taken from the set $\{24, 54, 84, 114, 114\}$, hence there are $5 \times 5 = 25$ injectors in total. Meanwhile, the horizontal and vertical coordinates of the producers are integers from the set $\{9, 39, 69, 99, 129, 159\}$, and there are $6 \times 6 = 36$ producers altogether.

In this case study, the unknown parameters are PERMX (in the natural logarithmic scale) on all gridblocks. Hence, the total number of parameters to be estimated is 27889. The phases in the reservoir include oil and water. Fig. 5(a) shows the reference PERMX map, which is used to generate observations in this case study. Specifically, the types of observations include well oil production rates (WOPR) and well water production rates (WWPR) from producers, and well bottom hole pressures (WBHP) and well water injector rates (WWIR) from injectors. The length of the data assimilation time window is 1440 days, with 9 evenly spanned reporting time steps. At each reporting time step, there are 122 data points. Therefore, the total number of observation data is $122 \times 9 = 1098$. As in the previous case study, a certain amount of zero-mean Gaussian white noise is introduced to the observation data. However, unlike the setting in the previous case study, here the noise STD is specified based on absolute measurement errors. For WWPR and WOPR, their noise STD is 8 m^3/day (around 0.9% of the maximum

Table 2
Data mismatch (DM), root mean squared error (RMSE) and ensemble spread in the M-5Spots case.

	Initial ensemble	Final ensemble (AutoAdaLoc)	Final ensemble (P-AutoAdaLoc-0)
DM (mean ± STD)	$(1.7091 \pm 0.3517) \times 10^7$	$(1.7896 \pm 0.5328) \times 10^5$	$\mathbf{(7.2535 \pm 1.1881) \times 10^4}$
RMSE (mean ± STD)	1.6647 ± 0.0707	1.3811 ± 0.0357	$\mathbf{1.2550 \pm 0.0340}$
Spread	1.1864	0.9293	**0.5819**

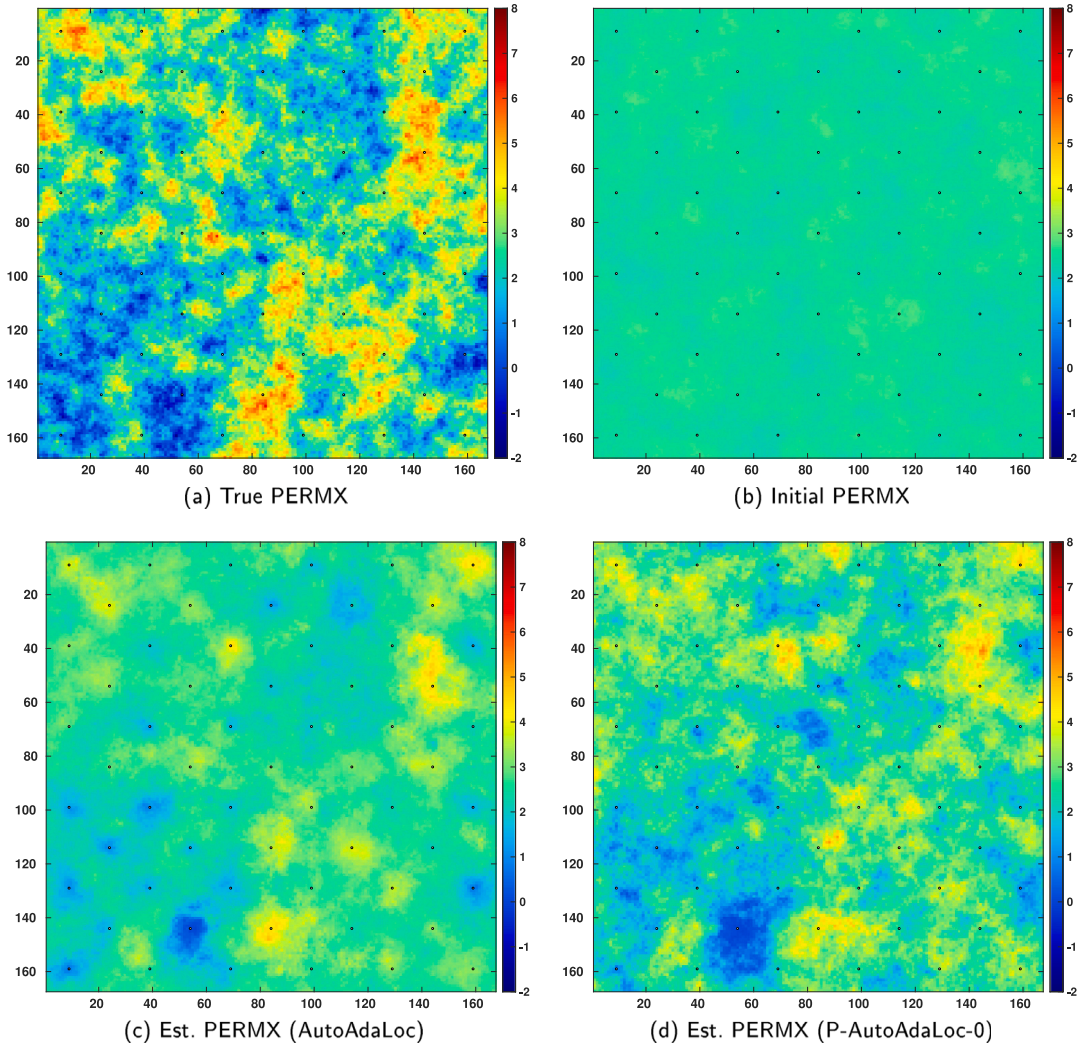


Fig. 5. PERMX maps (in the natural logarithmic scale) with respect to (a) the reference model; (b) the mean of the initial ensemble; (c) – (d) the means of the final estimated (est.) ensembles obtained by the IES algorithm with the AutoAdaLoc and the P-AutoAdaLoc-0 schemes, respectively, in the M-5Spots case. In all the maps, the small dots indicate the locations of wells.

value of WWPR and WOPR); for WBHP, the noise STD is 10 Psi (around 0.2% of the maximum value of WBHP); whereas for WWIR, the noise STD is 5 m^3/day (around 0.3% of the maximum value of WWIR).

The configuration of the IES algorithm in the M-5Spots case is roughly the same as that in the 5Spots case. Therefore, here we choose to only highlight some essential aspects of the algorithm configuration. Concretely, the IES algorithm is initialized with an ensemble of 100 reservoir models, with the mean PERMX map shown in Fig. 5(b). For the previously explained reason, here we let the IES algorithm be equipped with the AutoAdaLoc and the P-AutoAdaLoc-0 schemes, following Settings (2) and (4), respectively, in the 5Spots case. In the AutoAdaLoc scheme, the threshold values $\theta_{G_k,s}$ are pre-calculated according to Eq. (15) and then fixed afterward. Meanwhile, in the P-AutoAdaLoc-0 scheme, each reservoir model is associated with a set of 1098 localization length scales. An initial ensemble of localization length scales (with the ensemble size $N_e = 100$) is generated in the same way as in the 5Spots case. For illustration, the histograms of the initial ensembles of a few selected localization length scales are shown in the first column of Fig. 6. Other than the above aspects, the maximum number of iteration steps in the IES algorithm is also set to 20, and the stopping criteria in the M-5Spots are the same as those in the 5Spots case.

With the above experiment settings, the IES algorithm with the P-AutoAdaLoc-0 scheme outperforms that with the AutoAdaLoc scheme. To illustrate this point, Table 2 reports DM and RMSE (in terms of mean ± STD), as well as ensemble spread, with respect to the initial ensemble, and the final ones obtained by the IES algorithm with the AutoAdaLoc and P-AutoAdaLoc-0 schemes, respectively. In terms of mean DM, it can be seen that both localization schemes lead to significant (at least two orders of magnitude) reduction, in contrast to the initial ensemble without data assimilation. Comparing the AutoAdaLoc and P-AutoAdaLoc-0 schemes, the DM value with respect to

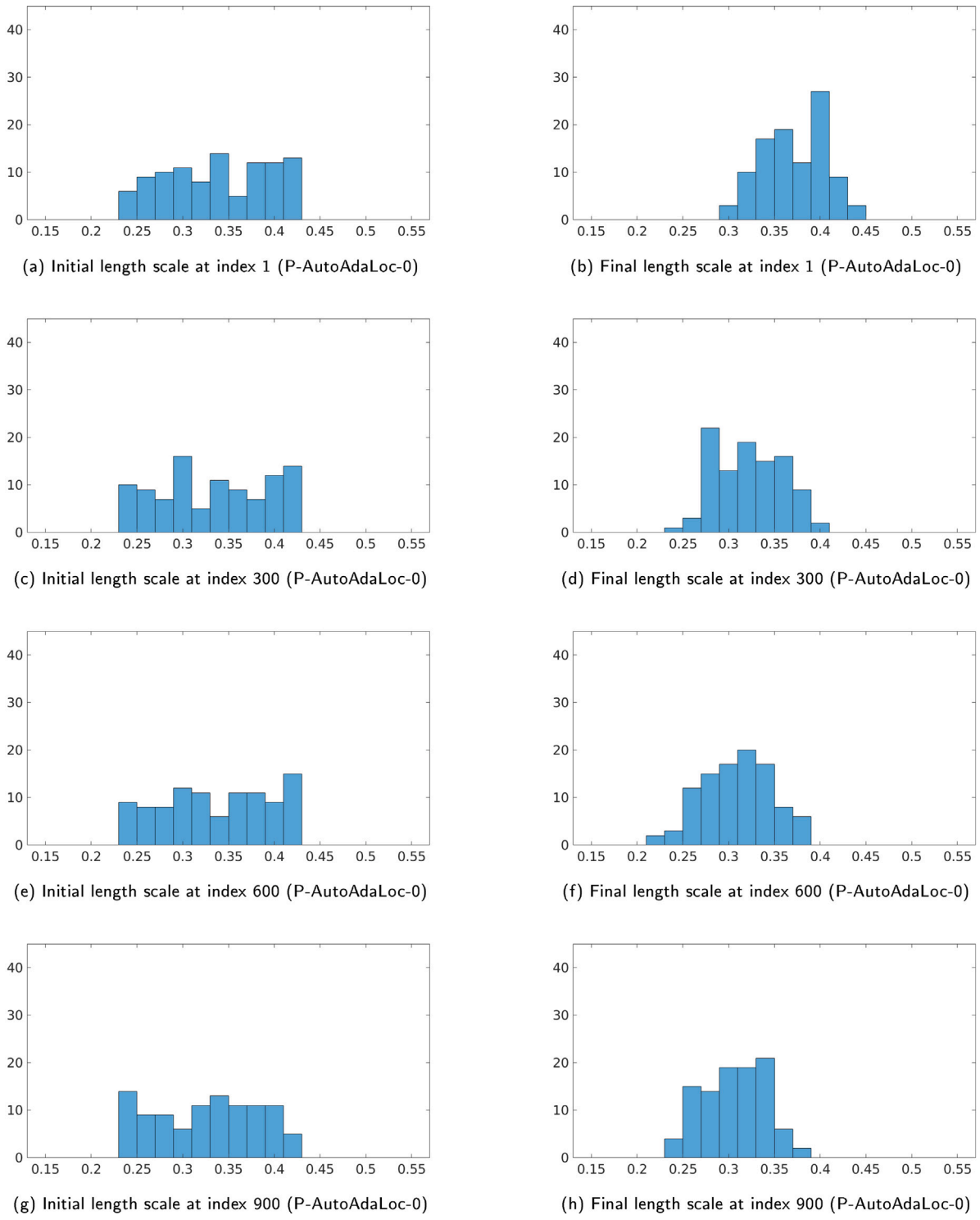


Fig. 6. Histograms of the initial (first column) and final (second column) ensembles of localization length scales at indices 1, 300, 600 and 900, respectively, with respect to the P-AutoAdaLoc-0 scheme used in the M-5Spots case.

the P-AutoAdaLoc-0 scheme is around 53.5% lower than that of the AutoAdaLoc scheme. In the meantime, the mean RMSE values obtained by the AutoAdaLoc and P-AutoAdaLoc-0 schemes are around 17.0% and 24.6%, respectively, lower than that of the initial ensemble. As for ensemble spread, both localization schemes result in reduced spread values. In this particular case, the P-AutoAdaLoc-0 scheme leads to a more substantial (about 51.0%) spread reduction than the AutoAdaLoc scheme does (about 21.7% reduction). Nevertheless, the P-AutoAdaLoc-0 scheme is still able to maintain sufficient ensemble spread and avoid ensemble collapse.

In addition, Fig. 7 presents the box plots of DM (first row) and RMSE (second row) at different iteration steps, when the AutoAdaLoc (first column) and the P-AutoAdaLoc-0 (second column) schemes, respectively, are applied to the IES algorithm. Conforming with the results

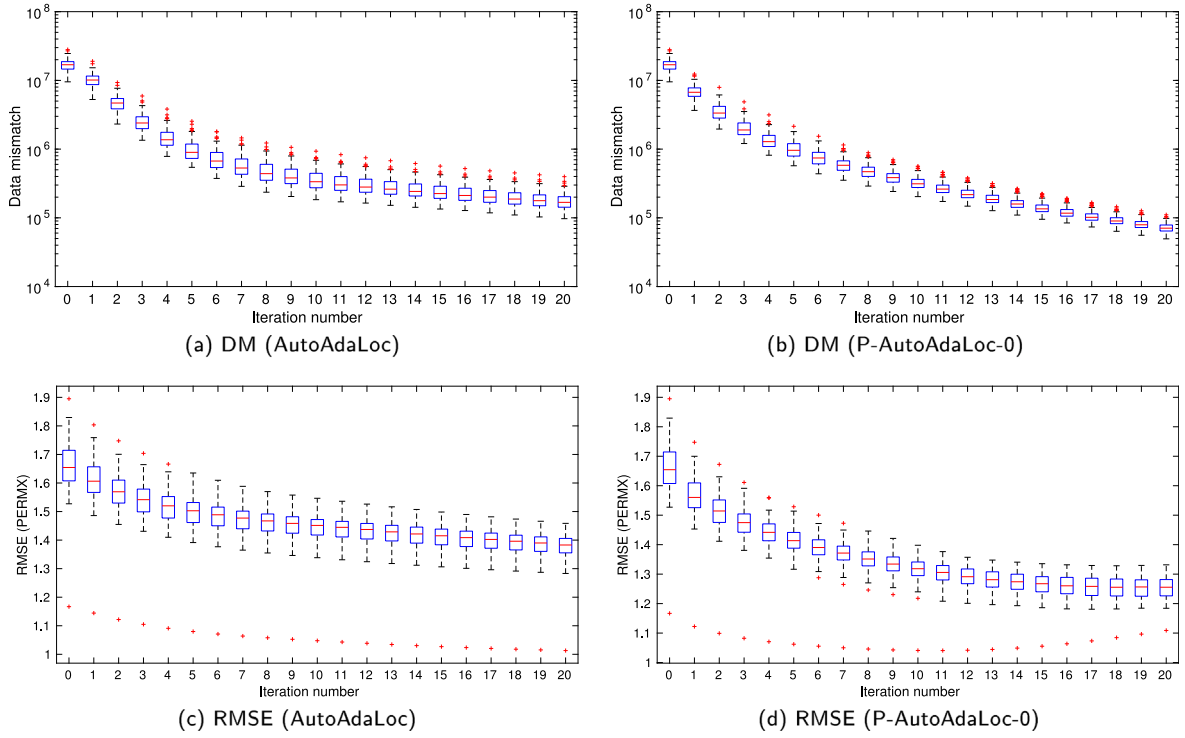


Fig. 7. Box plots of DM (first row) and RMSE of log PERMX (second row) at different iteration steps of the IES algorithm, which is equipped with the AutoAdaLoc (first column) and the P-AutoAdaLoc-0 (second column) schemes, respectively, in the M-5Spots case.

in Table 2, one can see that with the P-AutoAdaLoc-0 scheme, the corresponding DM and RMSE tend to converge faster to some lower values than those of the AutoAdaLoc scheme.

Fig. 5 displays the mean PERMX maps with respect to the initial ensemble (upper right), and the final ensembles delivered by the IES algorithm with the AutoAdaLoc (lower left) and the P-AutoAdaLoc-0 schemes (lower right), respectively. For comparison, the true PERMX map from the reference model is also included (upper left). Inspecting the initial mean and the true maps, it can be seen that the initial mean map is relatively flat, with fewer spatial variations. By assimilating production data into the reservoir models through the IES algorithm, the final mean maps with respect to both localization schemes exhibit more spatial variations, and can capture some of the geological features in the true map. A further comparison between the final mean maps of the two localization schemes helps reveal that in this particular case, the final mean map with respect to the AutoAdaLoc scheme appears spatially over-smoothed, whereas the final mean map obtained with the P-AutoAdaLoc-0 scheme is able to capture more geological features in the true map, and thus appears geologically more realistic than that obtained with the AutoAdaLoc scheme.

Finally, Fig. 6 plots the histograms of the initial (first column) and final (second column) ensembles of a few selected localization length scales used in the P-AutoAdaLoc-0 scheme. Since the initial ensemble of localization length scales contains samples drawn from a uniform distribution, the histograms of the initial ensemble appear relatively flat. As in Fig. 3, after data assimilation, the histograms of the final ensemble of localization length scales are more concentrated, but the changes of the histograms (in terms of locations, widths, and heights, etc.) appear less substantial, in comparison to the situation in Fig. 3(b) of the 5Spots case.

Overall, in this particular case study, one can conclude that in terms of DM and RMSE, the P-AutoAdaLoc-0 scheme performs better than the AutoAdaLoc scheme.

3.3. Case 3: Brugge benchmark

The last example considers the 3D Brugge benchmark case (Peters et al., 2010). The dimension of the reservoir model is $139 \times 48 \times 9$. There are 60048 reservoir gridblocks in total, and 44550 of them are active. The benchmark case provides an initial ensemble of 104 members. We use one of them as the reference model to generate observation data, and the rest to form the initial ensemble in data assimilation.

There are 20 producers and 10 water injectors in the field. The production period is 10 years with 20 report times. The types of production data used in data assimilation include well oil production rates (WOPR) and water cuts (WWCT) at 20 producers, and bottom hole pressures (WBHP) at all 30 wells. As such, the total number of production data is 1400. To mimic the situation in practical data assimilation problems, we introduce zero-mean Gaussian white noise to the production data generated by the reference model. Concretely, for WOPR and WWCT, their noise STDs are the maximum values between 10% of their magnitudes and 10^{-6} (which is adopted to avoid the potential numerical issue of division by zero); For WBHP data, their noise STDs are set to 1 bar.

The unknown parameters to be estimated include permeability along x-, y- and z-directions (PERMX, PERMY, PERMZ) in the natural logarithmic scale and porosity (PORO) on active gridblocks. As such, the total number of unknown parameters is $4 \times 44550 = 178200$. For illustration, the upper panels of Figs. 8 and 9 show the maps of PERMX and PORO, respectively, on Layer 2 of the reference model (upper left) and the mean reservoir model of the initial ensemble (upper right).

The configuration of the IES algorithm in the Brugge benchmark case is also very similar to those in the 5Spots and M-5Spots cases. For the purpose of performance comparison, we apply the AutoAdaLoc and the P-AutoAdaLoc-0 schemes to the IES algorithm. In the AutoAdaLoc scheme, the way of determining the threshold values $\theta_{G_k,s}$ is the same as that in the previous case studies. As for the P-AutoAdaLoc-0 scheme, each reservoir model is now associated with a set of 1400 localization

Table 3
Data mismatch (DM), root mean squared error (RMSE), and ensemble spread in the Brugge benchmark case.

	Initial ensemble	Final ensemble (AutoAdaLoc)	Final ensemble (P-AutoAdaLoc-0)
DM (mean ± STD)	$(0.3623 \pm 1.4900) \times 10^{10}$	$(0.9481 \pm 1.9730) \times 10^{7}$	$\mathbf{(3.9842 \pm 7.0173) \times 10^{5}}$
Total RMSE (mean ± STD)	1.5450 ± 0.3362	1.2610 ± 0.1663	$\mathbf{1.1645 \pm 0.1046}$
RMSE of PERMX (mean ± STD)	1.6585 ± 0.3827	1.3498 ± 0.1982	$\mathbf{1.2399 \pm 0.1205}$
RMSE of PERMY (mean ± STD)	1.6612 ± 0.3794	1.3546 ± 0.1959	$\mathbf{1.2444 \pm 0.1197}$
RMSE of PERMZ (mean ± STD)	2.0077 ± 0.4096	1.6426 ± 0.1937	$\mathbf{1.5282 \pm 0.1307}$
RMSE of PORO (mean ± STD)	0.0302 ± 0.0033	0.0298 ± 0.0031	$\mathbf{0.0259 \pm 0.0018}$
Spread	0.8661	0.6308	**0.5435**

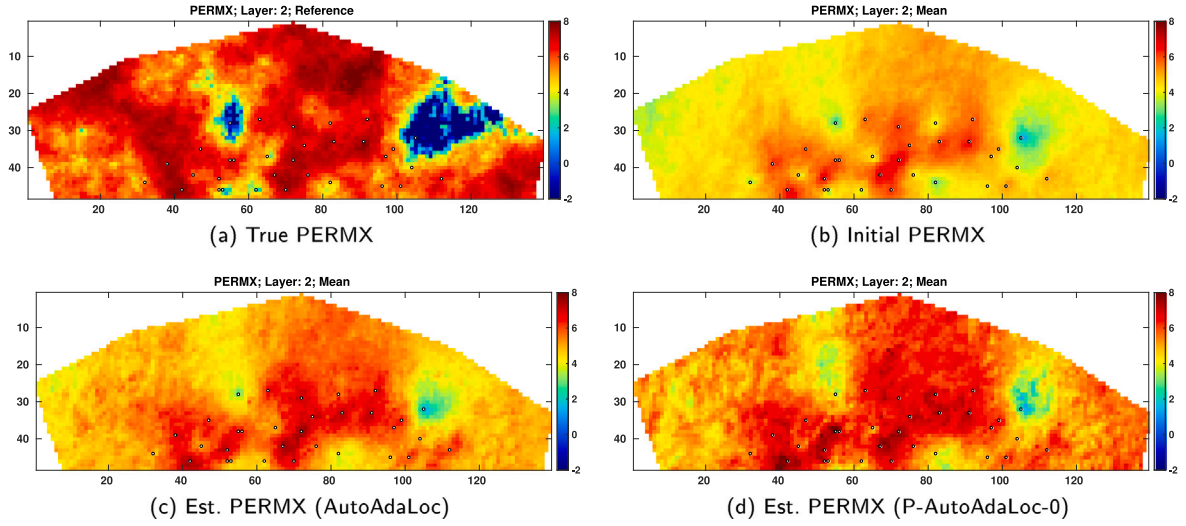


Fig. 8. PERMX maps (in the natural logarithmic scale) on Layer 2 of (a) the reference model; (b) the mean reservoir model of the initial ensemble; (c)–(d) the mean reservoir models of the final estimated (est.) ensembles obtained by the IES algorithm with the AutoAdaLoc and the P-AutoAdaLoc-0 schemes, respectively, in the Brugge benchmark case. In all the maps, the small dots indicate the locations of wells.

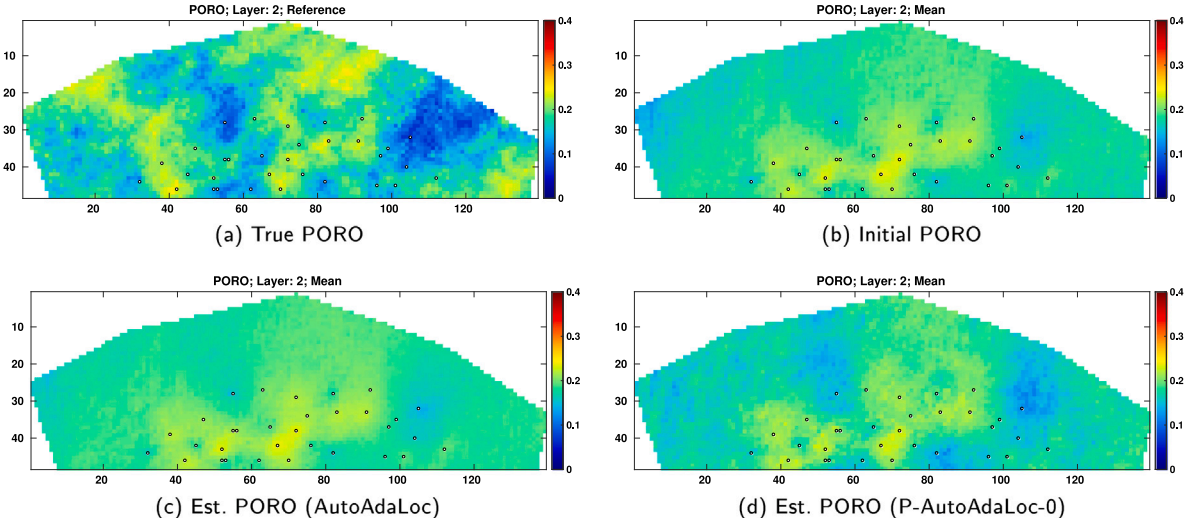


Fig. 9. As in Fig. 8, but for PORO map on Layer 2 of the Brugge benchmark case.

length scales. The localization length scales in the initial ensemble (now with the ensemble size $N_e = 103$) are again generated in the same way as that in the 5Spots case. For illustration, the first column of Fig. 10 displays the histograms of the initial ensembles of a few selected localization length scales. Meanwhile, the IES algorithm has maximum 20 iteration steps, and it shares the same stopping criteria as those in the 5Spots and M-5Spots case studies.

The numerical results below indicate that in this particular benchmark case, the P-AutoAdaLoc-0 scheme also tends to result in better performance than the AutoAdaLoc scheme. To support this point, Table 3 provides a summary of DM, RMSE, and ensemble spreads for the initial ensemble, and the final ensembles obtained by the IES algorithm with the AutoAdaLoc and the P-AutoAdaLoc-0 schemes, respectively. Coupled with either localization scheme, the IES algorithm is able to reduce the mean DM value for at least two orders of magnitude, in comparison to that of the initial ensemble, whereas the final mean DM of the P-AutoAdaLoc-0 scheme is at least one order of magnitude lower than that of the AutoAdaLoc scheme.

In terms of mean total RMSE (i.e., the RMSE with respect to all types of parameters to be estimated), the values of the IES algorithm with the AutoAdaLoc and P-AutoAdaLoc-0 schemes are 18.4% and 24.6%, respectively, lower than that of the initial ensemble, while

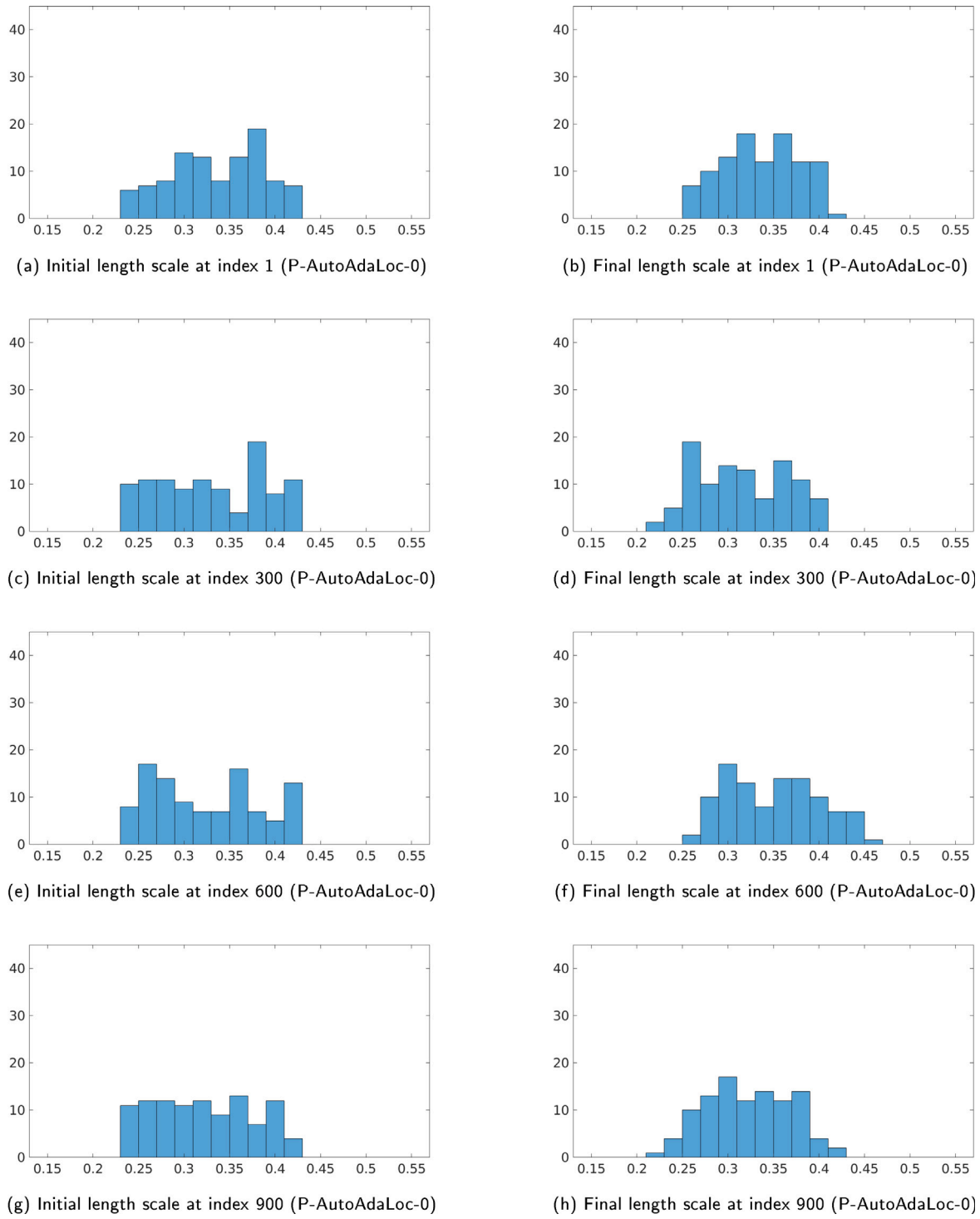


Fig. 10. Histograms of the initial (first column) and final (second column) ensembles of localization length scales at indices 1, 300, 600 and 900, respectively, with respect to the P-AutoAdaLoc-0 scheme used in the Brugge benchmark case.

the P-AutoAdaLoc-0 scheme achieves around 6.2% lower mean total RMSE than the AutoAdaLoc scheme does. Similar results can also be observed for RMSE values with respect to individual types of petrophysical parameters (PERMX, PERMY, PERMZ, and PORO). Finally, regarding ensemble spread, one can see that the values obtained by the AutoAdaLoc and the P-AutoAdaLoc-0 schemes are 27.2% and 37.2%, respectively, lower than that of the initial ensemble. The ensemble spread of the P-AutoAdaLoc-0 scheme is around 10.0% lower than that of the AutoAdaLoc scheme, but is substantially away from zero, meaning that ensemble collapse is avoided.

Supplementary to the results in Table 3, Fig. 11 displays the box plots of DM (first row), total RMSE (second row), RMSEs of PERMX (third row) and PORO (fourth row) at different iterations of the IES algorithm, when the AutoAdaLoc (first column) or the P-AutoAdaLoc-0 (second column) scheme is adopted. In all these box plots, it is clear that the P-AutoAdaLoc-0 scheme tends to achieve final DM or RMSE values lower than those of the AutoAdaLoc scheme.

For further illustration, Figs. 8 and 9 present PERMX and PORO maps on Layer 2 of the reference model (upper left), the mean reservoir model of the initial ensemble (upper right) and the mean reservoir

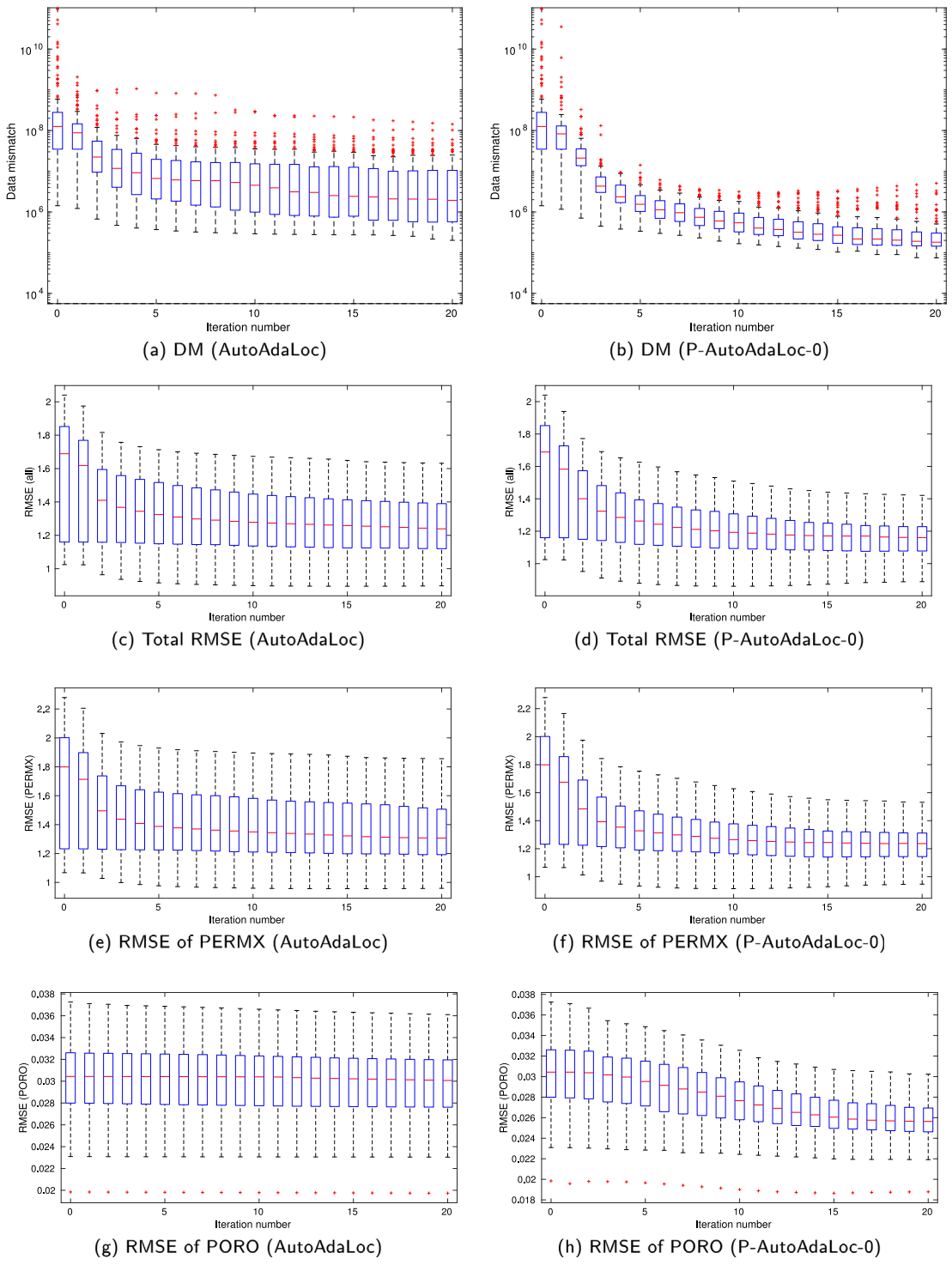


Fig. 11. Box plots of DM (first row) and RMSE of log PERMX (second row) at different iteration steps of the IES algorithm, which is equipped with the AutoAdaLoc (first column) and the P-AutoAdaLoc-0 (second column) schemes, respectively, in the Brugge benchmark case.

models of the final ensembles obtained by the IES algorithm with the AutoAdaLoc scheme (lower left) and the P-AutoAdaLoc-0 scheme (lower right), respectively. A visual comparison among these maps reveals that the estimated PERMX and PORO maps tend to be more similar to the reference ones, in contrast to the corresponding maps of the initial mean reservoir model. Moreover, the maps obtained with the P-AutoAdaLoc-0 scheme resemble the reference maps better than those with the AutoAdaLoc scheme, which is particularly evident for the PORO maps.

Fig. 10 also demonstrates the prior and posterior distributions of a few selected localization length scales in the P-AutoAdaLoc-0 scheme. Similar to the situations in the 5Spots and M-5Spots cases, there are some noticeable differences between the histograms of the initial and final ensembles of these localization length scales, but such changes can be considered moderate in comparison to the situation in the first row

of Fig. 3, where only a single localization length scale is adopted in the P-AutoAdaLoc-0 scheme (cf Setting (3) of the 5Spots case).

4. Discussion and conclusion

The focus of the current work is on further improving the performance of a correlation-based, automatic and adaptive localization (AutoAdaLoc) scheme used in an iterative ensemble smoother (IES), by introducing a continuous hyper-parameter optimization (CHOP) procedure to optimize the values of localization length scales in a parameterized AutoAdaLoc (P-AutoAdaLoc) scheme. The CHOP procedure converts a hyper-parameter optimization problem into a conventional parameter estimation problem, and solves it through an IES algorithm. In doing so, the CHOP procedure possesses some practical benefits inherent in ensemble-based data assimilation algorithms, including the nature of being derivative-free and the ability to provide uncertainty quantification for the estimated quantities to some extent. As such, in comparison to the existing distance- or correlation-based localization schemes, there are some noticeable differences in the proposed P-AutoAdaLoc scheme: Instead of specifying static, point values for localization length scales, the proposed scheme iteratively estimates an ensemble of localization length scales, and can be naturally integrated into the IES-based data assimilation workflow. Moreover, like the ability of the IES algorithm in dealing with large-scale reservoir data assimilation problems, the P-AutoAdaLoc scheme has the capacity to handle a relatively large number of localization length scales. As such, the P-AutoAdaLoc scheme can surpass the limitation in many existing localization schemes that only contain a single hyper-parameter, in order to avoid the complexity of simultaneously optimizing multiple hyper-parameters. The performance of the P-AutoAdaLoc scheme is investigated in three synthetic case studies, in which the numerical results indicate that the P-AutoAdaLoc scheme indeed tends to perform better than the original AutoAdaLoc scheme.

As indicated in Algorithm 2, at a given iteration step, the IES algorithm with the P-AutoAdaLoc scheme conducts the same number of forward reservoir simulations as that with the AutoAdaLoc scheme, which is a desirable feature. Meanwhile, the presence of the CHOP procedure does incur additional algorithmic complexity and computational overheads. The main cause of these overheads lies in the fact that in the P-AutoAdaLoc scheme, each reservoir model is associated with an individual set of multiple localization length scales, making the model update formula (e.g., Eq. (17)) depend on the individual combination of a reservoir model and a set of localization length scales. This means that a tapering matrix needs to be calculated for each reservoir model, which could consume a large amount of computer memory with both big models and big field datasets. For the three case studies in the current work, since the number of production data is relatively small, this noticed issue does not appear to cause any problem. Nevertheless, should a large number of field datasets, e.g., 4D seismic (Lorentzen et al., 2019; Luo et al., 2017; Soares et al., 2020), be assimilated into a big reservoir model, then it could be computationally challenging to construct a big tapering matrix for the update of each reservoir model. In this regard, we expect that a few strategies can be adopted to mitigate the consumption of computer memory, which include: (1) sparse model/data representation (Canchumuni et al., 2019; Lorentzen et al., 2019; Luo et al., 2017; Soares et al., 2020) to reduce the size(s) of reservoir model and/or observation data; (2) projection of observation data onto the ensemble subspace and then using the projected data as the effective observations (Luo et al., 2019; Luo and Cruz, 2022); (3) local analysis in which each update focuses on a small group of model variables and observation data points (Chen and Oliver, 2017; Soares et al., 2021). In future work, we will test some of these strategies in relevant data assimilation problems.

In the current work, we have confined ourselves to the tuning of localization length scales through the CHOP procedure. At a more general level, we expect that the CHOP procedure may be similarly applied to optimize hyper-parameters in various data assimilation algorithms, as exemplified in the preceding work (Luo and Xia, 2022). More investigations in this regard are thus expected in our future work.

Nomenclature

c	=	scalar coefficient
$\mathbf{C}_d$	=	covariance matrix of data noise
$\mathbf{C}_m^i$	=	sample error covariance matrix of reservoir models at the ith iteration
$\mathbf{C}_{\ell}^i$	=	sample error covariance matrix of hyper-parameters at the ith iteration
C_j^{i+1}	=	the jth cost function at the $(i+1)$ iteration step in a minimum-average-cost (MAC) problem
$\tilde{C}_j^{i+1}$	=	similar to C_j^{i+1}, but for the MAC problem of optimizing hyper-parameters
$\mathbf{d}^o$	=	noisy observed data
$\mathbf{d}_j^o$	=	the jth perturbation of $\mathbf{d}^o$
$\tilde{\mathbf{d}}_j^o$	=	$\mathbf{d}^o$ normalized by a square root matrix of $\mathbf{C}_d$
$\mathbf{d}^{sim}$	=	simulated observations
$\mathbf{f}$	=	vector mapping that represents a model update formula
$\tilde{\mathbf{f}}$	=	shorter notation of $\mathbf{f}$ after dropping some invariants
f_{GC}	=	Gaspari–Cohn function
$\mathbf{g}$	=	reservoir forward simulator
$\tilde{\mathbf{g}}$	=	$\mathbf{g}$ normalized by a square root matrix of $\mathbf{C}_d$
G_k	=	the group (e.g., permeability, porosity) of petro-physical parameters that $m_{j,k}^i$ belongs to
$\mathbf{h}$	=	composition of the functions of $\tilde{\mathbf{f}}$ and $\mathbf{g}$
$\mathbf{I}_p$	=	the p-dimensional identity matrix
$k_{k,s}^i$	=	matrix entry at the kth row and the sth column of $\mathbf{K}^i$
$\mathbf{K}^i$	=	Kalman-gain-like matrix at the ith iteration step for updating reservoir models
$\mathbf{K}_{\ell}^i$	=	Kalman-gain-like matrix at the ith iteration step for updating hyper-parameters
$\mathbf{L}^i$	=	an ensemble containing the elements $\boldsymbol{\ell}_j^i$ for $j = 1, \cdots, N_e$
m	=	dimension of a reservoir model
$m_{j,k}^i$	=	the kth model variable of $\mathbf{m}_j^i$
$\mathbf{m}$	=	generic reservoir model
$\mathbf{m}^a$	=	analysis (posterior) reservoir model
$\mathbf{m}^b$	=	background (prior) reservoir model
$\mathbf{m}_j^i$	=	the jth reservoir model at the ith iteration step
$\mathbf{M}^i$	=	the ensemble of reservoir models at the ith iteration step
$\bar{\mathbf{m}}^i$	=	mean reservoir model of $\mathbf{M}^i$
$\mathbf{m}^{ref}$	=	reference reservoir model
N_e	=	ensemble size
p	=	dimension of the observation vector
$\mathbf{S}_{\ell}^i$	=	hyper-parameter square-root matrix at the ith iteration step
$\mathbf{S}_g^i$	=	data square-root matrix at the ith iteration step
$\tilde{\mathbf{S}}_g^i$	=	$\mathbf{S}_g^i$ normalized by a square root matrix of $\mathbf{C}_d$
$\mathbf{S}_h^i$	=	data square-root matrix at the ith iteration step based on the composition function $\mathbf{h}$
$\mathbf{S}_m^i$	=	model square-root matrix at the ith iteration step
$\mathbf{S}$	=	vector containing ensemble standard deviations for all model variables
$t_{k,s}^i$	=	matrix entry at the kth row and the sth column of $\mathbf{T}^i$
$t_{j,k,s}^i$	=	$t_{k,s}^i$ associated with the jth reservoir model
$\mathbf{T}^i$	=	tapering matrix at the ith iteration step
$\mathbf{T}\left(\boldsymbol{\ell}_j^i\right)$	=	tapering matrix as a function of $\boldsymbol{\ell}_j^i$ for updating the reservoir model $\mathbf{m}_j^i$
$\mathbf{T}_{\ell}\left(\boldsymbol{\ell}_j^i\right)$	=	similar to $\mathbf{T}\left(\boldsymbol{\ell}_j^i\right)$, but for updating the hyper-parameter vector $\boldsymbol{\ell}_j^i$.

$\mathbf{x}$	=	dummy vector
$[\mathbf{x}]_s$	=	the sth element of $\mathbf{x}$
ℓ_s	=	localization length scale associated with the sth observation
$\ell_{k,s}$	=	localization length scale associated with the kth model variable and the sth observation
$\ell^i_{j,s}$	=	the sth element of $\boldsymbol{\ell}^i_j$
$\boldsymbol{\ell}^i_j$	=	a set of localization length scales associated with $\mathbf{m}^i_j$
$\bar{\boldsymbol{\ell}}^i$	=	mean hyper-parameter vector with respect to the ensemble $\mathbf{L}^i$
$\boldsymbol{\epsilon}^0_{G_k,s}$	=	substitute sampling errors in the correlation field $\boldsymbol{\rho}^0_{G_k,s}$
γ^i	=	regularization parameter at the ith iteration step
$\rho^0_{k,s}$	=	sample correlation coefficient between the ensembles $\{m^0_{j,k}\}_{j=1}^{N_e}$ and $\left\{[\tilde{\mathbf{g}}\left(\mathbf{m}^0_j\right)]_s\right\}_{j=1}^{N_e}$
$\boldsymbol{\rho}^0_{G_k,s}$	=	sample correlation field between $\left\{[\tilde{\mathbf{g}}\left(\mathbf{m}^0_j\right)]_s\right\}_{j=1}^{N_e}$ and the group G_k of petro-physical parameters
$\sigma_{G_k,s}$	=	estimated noise level in the sample correlation field $\boldsymbol{\rho}^0_{G_k,s}$
σ_k	=	ensemble standard deviation with respect to the kth model variable
$\theta_{G_k,s}$	=	threshold value for the group G_k of petro-physical parameters
$\Delta\tilde{d}^i_{j,s}$	=	the sth element of $\Delta\tilde{\mathbf{d}}^i_j$
$\Delta\tilde{\mathbf{d}}^i_j$	=	normalized innovation defined as $\tilde{\mathbf{d}}^o_j - \tilde{\mathbf{g}}\left(\mathbf{m}^i_j\right)$

Subscripts

e	=	ensemble
j	=	index of ensemble member
k	=	index of model variable
m	=	model
ref	=	reference
s	=	index of observation element
sim	=	simulation

Superscripts

a	=	analysis
b	=	background
i	=	index of iteration step
o	=	observation

CRediT authorship contribution statement

Xiaodong Luo: Conceptualization, Methodology, Software, Numerical investigation, Writing, Reviewing. **William C. Cruz:** Methodology, Software, Numerical investigation. **Xin-Lei Zhang:** Methodology, Writing, Reviewing. **Heng Xiao:** Methodology, Writing, Reviewing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

The authors would like to thank the reviewers for their constructive and valuable suggestions, which helped improve the quality of this work. XL acknowledges partial financial support from both the NORCE research project "Assimilating 4D Seismic Data: Big Data Into Big Models" which is funded by industry partners, Equinor Energy AS, Lundin Energy Norway AS, Repsol Norge AS, Shell Global Solutions International B.V., TotalEnergies EP Norge AS, and Wintershall Dea Norge AS, as well as the Research Council of Norway (project number: 295002), and the National Centre for Sustainable Subsurface Utilization of the Norwegian Continental Shelf (NCS2030), which is funded by industry partners and the Research Council of Norway (project number: 331644). 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. The authors would also like to thank SLB for providing academic licenses to ECLIPSE′.

References

Aanonsen, S., Nævdal, G., Oliver, D., Reynolds, A., Vallès, B., 2009. The ensemble Kalman filter in reservoir engineering: a review. SPE J. 14, 393–412, SPE-117274-PA.

Anderson, J.L., 2009. Spatially and temporally varying adaptive covariance inflation for ensemble filters. Tellus 61A, 72–83.

Anderson, J.L., 2012. Localization and sampling error correction in ensemble Kalman filter data assimilation. Mon. Weather Rev. 140 (7), 2359–2371.

Anderson, J., Lei, L., 2013. Empirical localization of observation impact in ensemble Kalman filters. Mon. Weather Rev. 141 (11), 4140–4153.

Arroyo, E., Devegowda, D., Datta-Gupta, A., Choe, J., 2008. Streamline-assisted ensemble Kalman filter for rapid and continuous reservoir model updating. SPE Reserv. Eval. Eng. 11, 1046–1060, SPE-104255-PA.

Bishop, C.H., Hodyss, D., 2007. Flow-adaptive moderation of spurious ensemble correlations and its use in ensemble-based data assimilation. Q. J. R. Meteorol. Soc. 133, 2029–2044.

Canchumuni, S.W., Emerick, A.A., Pacheco, M.A.C., 2019. History matching geological facies models based on ensemble smoother and deep generative models. J. Pet. Sci. Eng. 177, 941–958.

Chen, Y., Oliver, D.S., 2010. Cross-covariances and localization for EnKF in multiphase flow data assimilation. Comput. Geosci. 14, 579–601.

Chen, Y., Oliver, D., 2013. Levenberg–Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification. Comput. Geosci. 17, 689–703.

Chen, Y., Oliver, D.S., 2017. Localization and regularization for iterative ensemble smoothers. Comput. Geosci. 21, 13–30.

Donoho, D.L., Johnstone, J.M., 1994. Ideal spatial adaptation by wavelet shrinkage. Biometrika 81, 425–455.

Donoho, D.L., Johnstone, I.M., 1995. Adapting to unknown smoothness via wavelet shrinkage. J. Amer. Statist. Assoc. 90, 1200–1224.

Emerick, A., Reynolds, A., 2011. Combining sensitivities and prior information for covariance localization in the ensemble Kalman filter for petroleum reservoir applications. Comput. Geosci. 15, 251–269.

Emerick, A.A., Reynolds, A.C., 2012. Ensemble smoother with multiple data assimilation. Comput. Geosci. 55, 3–15.

Evensen, G., 2009. Data Assimilation: The Ensemble Kalman Filter. Springer Science & Business Media.

Evensen, G., Raanes, P.N., Stordal, A.S., Hove, J., 2019. Efficient implementation of an iterative ensemble smoother for big-data assimilation and reservoir history matching. Front. Appl. Math. Stat. 5, 47.

Fertig, E.J., Hunt, B.R., Ott, E., Szunyogh, I., 2007. Assimilating non-local observations with a local ensemble Kalman filter. Tellus A 59, 719–730.

Furrer, R., Bengtsson, T., 2007. Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants. J. Multivariate Anal. 98, 227–255.

Gaspari, G., Cohn, S.E., 1999. Construction of correlation functions in two and three dimensions. Q. J. R. Meteorol. Soc. 125, 723–757.

Hamill, T.M., Whitaker, J.S., Anderson, J.L., Snyder, C., 2009. Comments on "sigma-point Kalman filter data assimilation methods for strongly nonlinear systems". J. Atmos. Sci. 66, 3498–3500.

Hamill, T.M., Whitaker, J.S., Snyder, C., 2001. Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. Mon. Wea. Rev. 129, 2776–2790.

Lacerda, J.M., Emerick, A.A., Pires, A.P., 2019. Methods to mitigate loss of variance due to sampling errors in ensemble data assimilation with non-local model parameters. J. Pet. Sci. Eng. 172, 690–706.

Lorentzen, R., Bhakta, T., Grana, D., Luo, X., Valestrand, R., Nævdal, G., 2020. Simultaneous assimilation of production and seismic data: Application to the Norne field. Comput. Geosci. 24, 907–920. http://dx.doi.org/10.1007/s10596-019-09900-0.

Lorentzen, R., Luo, X., Bhakta, T., Valestrand, R., 2019. History matching the full Norne field model using seismic and production data. SPE J. 24, 1452–1467, SPE-194205-PA.

Luo, X., 2021. Novel iterative ensemble smoothers derived from a class of generalized cost functions. Comput. Geosci. 25, 1159–1189.

Luo, X., Bhakta, T., 2020. Automatic and adaptive localization for ensemble-based history matching. J. Pet. Sci. Eng. 184, 106559.

Luo, X., Bhakta, T., Jakobsen, M., Nævdal, G., 2017. An ensemble 4D-seismic history-matching framework with sparse representation based on wavelet multiresolution analysis. SPE J. 22, 985–1010. http://dx.doi.org/10.2118/180025-PA, SPE-180025-PA.

Luo, X., Bhakta, T., Nævdal, G., 2018. Correlation-based adaptive localization with applications to ensemble-based 4D seismic history matching. SPE J. 23, 396–427. http://dx.doi.org/10.2118/185936-PA, SPE-185936-PA.

Luo, X., Cruz, W.C., 2022. Data assimilation with soft constraints (DASC) through a generalized iterative ensemble smoother. Comput. Geosci. 26 (3), 571–594.

Luo, X., Lorentzen, R.J., Valestrand, R., Evensen, G., 2019. Correlation-based adaptive localization for ensemble-based history matching: Applied to the Norne field case study. SPE Reserv. Eval. Eng. 22, 1084–1109. http://dx.doi.org/10.2118/191305-PA, SPE-191305-PA.

Luo, X., Stordal, A., Lorentzen, R., Nævdal, G., 2015. Iterative ensemble smoother as an approximate solution to a regularized minimum-average-cost problem: theory and applications. SPE J. 20, 962–982. http://dx.doi.org/10.2118/176023-PA, SPE-176023-PA.

Luo, X., Xia, C.A., 2022. Continuous Hyper-parameter OPtimization (CHOP) in an ensemble Kalman filter. Front. Appl. Math. Stat. 8, 1021551.

Nævdal, G., Johnsen, L.M., Aanonsen, S.I., Vefring, E.H., 2005. Reservoir monitoring and continuous model updating using ensemble Kalman filter. SPE J. 10, 66–74, SPE-84372-PA.

Oliver, D., Chen, Y., 2010. Recent progress on reservoir history matching: a review. Comput. Geosci. 15, 185–221.

Peters, L., Arts, R., Brouwer, G., Geel, C., Cullick, S., Lorentzen, R.J., Chen, Y., Dunlop, N., Vossepoel, F.C., Xu, R., et al., 2010. Results of the Brugge benchmark study for flooding optimization and history matching. SPE Reserv. Eval. Eng. 13, 391–405.

Ranazzi, P.H., Luo, X., Sampaio, M.A., 2022. Improving pseudo-optimal Kalman-gain localization using the random shuffle method. J. Pet. Sci. Eng. 215, 110589.

Skjervheim, J.A., Evensen, G., 2011. An ensemble smoother for assisted history matching. In: SPE Reservoir Simulation Symposium. The Woodlands, Texas, USA, SPE-141929-MS.

Soares, R., Luo, X., Evensen, G., Bhakta, T., 2020. 4D seismic history matching: Assessing the use of a dictionary learning based sparse representation method. J. Pet. Sci. Eng. 195, 107763.

Soares, R.V., Luo, X., Evensen, G., Bhakta, T., 2021. Handling big models and big data sets in history-matching problems through an adaptive local analysis scheme. SPE J. 26 (02), 973–992.

Van Leeuwen, P.J., Evensen, G., 1996. Data assimilation and inverse methods in terms of a probabilistic formulation. Mon. Wea. Rev. 124, 2898–2913.

Yu, T., Zhu, H., 2020. Hyper-parameter optimization: A review of algorithms and applications. arXiv preprint arXiv:2003.05689.