TU Delft

Delft University of Technology

Sequential model identification with reversible jump ensemble data assimilation method

Huan, Yue; Lin, Hai Xiang

DOI
10.1007/s11222-024-10499-1

Publication date
2024

Document Version
Final published version

Published in
Statistics and Computing

Citation (APA)
Huan, Y., & Lin, H. X. (2024). Sequential model identification with reversible jump ensemble data assimilation method. *Statistics and Computing*, *34*(6), Article 184. https://doi.org/10.1007/s11222-024-10499-1

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Statistics and Computing (2024) 34:184
https://doi.org/10.1007/s11222-024-10499-1

ORIGINAL PAPER

Sequential model identification with reversible jump ensemble data assimilation method

Yue Huan[1] · **Hai Xiang Lin**[1]

Received: 29 September 2023 / Accepted: 7 September 2024
© The Author(s) 2024

Abstract
In data assimilation (DA) schemes, the form representing the processes in the evolution models are pre-determined except some parameters to be estimated. In some applications, such as the contaminant solute transport model and the gas reservoir model, the modes in the equations within the evolution model cannot be predetermined from the outset and may change with the time. We propose a framework of sequential DA method named Reversible Jump Ensemble Filter (RJEnF) to identify the governing modes of the evolution model over time. The main idea is to introduce the Reversible Jump Markov Chain Monte Carlo (RJMCMC) method to the DA schemes to fit the situation where the modes of the evolution model are unknown and the dimension of the parameters is changing. Our framework allows us to identify the modes in the evolution model and their changes, as well as estimate the parameters and states of the dynamic system. Numerical experiments are conducted and the results show that our framework can effectively identify the underlying evolution models and increase the predictive accuracy of DA methods.

Keywords Data assimilation · State-space models · Bayesian inference · RJMCMC · Model identification

1 Introduction

The quality of a forecast depends on the accuracy of the initial condition, the dynamic system, and its dynamical consistency (Barthélémy et al. 2022). Data assimilation (DA) methods estimate the state based on observations, a dynamic system, and statistical information. DA has been successfully applied to improving the forecast accuracy in a wide range of dynamical models including ocean (van Leeuwen 2003; Vetra-Carvalho et al. 2018), weather forecast (Bachmann et al. 2020), air quality (Jin et al. 2019), and storm surge models (Wang et al. 2022).

From a statistical perspective, DA is equivalent to filtering inference in a state-space model. In a state-space model, the evolution models describe the dynamics of the state variables over time. In many applications, the evolution models which contain unknown processes and uncertain parameters are imperfect (Chang and Zhang 2019). The imperfection stems from an incompleteness, an inability to account for all relevant processes (Lewis et al. 2006).

Identifying the evolution equations corresponding to the actual physical processes is important, which can be interpreted as learning the modes and parameters of the evolution equations. In this study, we consider the PDE of the following form

$$\frac{\partial x_k}{\partial t} = \Phi(x_k, \alpha)\beta = \sum_{j=1}^{n_{cand}} \phi_{j,k}(x_k, \alpha_{j,k})\beta_{j,k}, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is an n-dimensional vector representing the state at time k. $k \in \mathbb{N}$ is a discrete time step. $\Phi(x, \alpha)$ denotes the library of n_{cand} candidate processes and empirical models, in which α are the unknown parameters, and β denotes the unknown coefficients. Let $\theta_k \triangleq \{\alpha_{1,k}^T, \alpha_{2,k}^T, \cdots, \alpha_{n_{cand},K}^T; \beta_{1,k}, \beta_{2,k}, \cdots, \beta_{n_{cand},k}\}^T$ be the vector of parameters at time k, where the dimensions of $\alpha_{j,k}$ can be different from each other and are denoted as $p_{\alpha,j,k}$, $j = 1, 2, \cdots, n_{cand}$. In some occasions, $\phi_i(x, \alpha_i)$ could not appear at the same time.

For example, for contaminant solute transport in subsurface formation, simultaneous processes may exist, such as

✉ Yue Huan
Y.Huan@tudelft.nl

Hai Xiang Lin
H.X.Lin@tudelft.nl

[1] Delft Institute of Applied Mathematics (DIAM), Delft University of Technology, 2628 CD Delft, The Netherlands

Published online: 18 September 2024

advection (ADV), dispersion (DIS), and sorption (SORP) (Chang and Zhang 2019). Different from the ADV and DIS, the SORP cannot be easily modeled (Chang and Zhang 2019). Various empirical models are usually proposed (based on laboratory experiments) for modeling SORP by considering different conditions. Two equilibrium sorption modes, such as Freundlich sorption isotherm (F-SORP) and Langmuir sorption isotherm (L-SORP), could occur in contaminant solute transport in subsurface formation. The processes F-SORP and L-SORP cannot happen concurrently. In the field of the gas reservoir, the identification of the evolution model is difficult because of the unknown underground processes and terrain (Chang and Zhang 2019; Lim et al. 2020).

Although there is an enormous literature on state estimation and sequential estimation of both states and parameters (Vrugt et al. 2013; Moradkhani et al. 2012; Stroud et al. 2018; Katzfuss et al. 2020; Drovandi et al. 2022), we have not found papers considering both the processes identification and the state and parameter estimation together. As opposite to off-line model identification, incorporating the model identification into the process of DA can enhance the precision of the evolution model and enable the identification of the change in the dynamics. So, this problem is worthy of study.

Within a Bayesian prospect, all relevant information about the states of a process $\{x_1, x_2, \cdots, x_k\}$ given observations up to and including time k can be obtained from the posterior distribution $p(x_1, x_2, \cdots, x_k|z_1, \cdots, z_k)$. $z_k \in \mathbb{R}^n$ is a m-dimensional vector representing the observation at time k. In the existing works, the model processes in Eq. (1) assumed to be known, which means that the modes in the equations within the evolution model are already decided. However, the evolution model cannot fit the underlying dynamical system perfectly in some situations. The modes in the evolution equation corresponding to the processes may change, which means the mode of the dynamic system changes, and also the type and the number of parameters may change during the time interval considered. So we cannot know when the dynamical system might change from one mode to another mode. For this reason, in the process of DA, the dynamic identification of underlying physical processes with the incorporation of observational data exhibits a diverse range of applications. What's more, this methodology can tackle the intricate issue of elucidating complex evolutionary equations during the preliminary phases of DA.

Typically, the DA problem is solved sequentially over a sequence of assimilation time windows, which are usually fixed-length intervals in time. Let $M : \mathbb{R}^n \to \mathbb{R}^n$ denote a mapping of the state space into itself. It is assumed that the state of the dynamic system evolves according to the nonlinear difference equation

$$x_k = M(x_{k-1}) + \epsilon_{k-1}, \tag{2}$$

where ϵ_{k-1} is the $n-$dimensional vector denoting the external forcing. It is usually assumed that ϵ_{k-1} is a white noise sequence (Lewis et al. 2006). $E(\epsilon_{k-1}) = 0$. It is serially uncorrelated, that is, $E(\epsilon_{k-1}\epsilon_{r-1}) = 0$ for $r \neq k$ and $Cov(\epsilon_{k-1}) = E(\epsilon_{k-1}{\epsilon_{k-1}}^T) = Q_{k-1} \in \mathbb{R}^{n\times n}$, is a known symmetric and positive definite matrix.

$h : \mathbb{R}^n \to \mathbb{R}^m$ is a mapping from the model space, $\mathbb{R}^n$ to the observation space, $\mathbb{R}^m$. Then

$$z_k = h(x_k) + v_k \tag{3}$$

defines in general, a nonlinear relationship between the observations z and the state x. $v_k \in \mathbb{R}^m$ is a white noise sequence with $E(v_k) = 0$ and $Cov(v_k) = R_k \in \mathbb{R}^{m\times m}$ and R_k is a real symmetric and positive definite matrix (Lewis et al. 2006).

Let x_0 be the random initial condition. In this framework, hierarchical state-space models (HSSMs) can be used as the general representations of systems observed over time. A HSSM describes the temporal evolution of the system state, and the relationship of the state x_k to observations z_k, which allows for unknown time-varying parameters in any part of the model. Here we focus on parameters θ included in evolution function $M(\cdot|\theta)$. In this case, the HSSM is given by

$$\begin{aligned} &Observation: z_k|x_k, \sim \quad \mathcal{N}_{m_k}(h_k(x_k), R_k), \\ &Evolution: x_k|x_{k-1}, \theta_k \sim \\ &\mathcal{N}_n(M_{k-1}(x_{k-1}|\theta_{k-1}), Q_{k-1}), \\ &Parameter: \theta_k|\theta_{k-1} \sim \quad p(\theta_k|\theta_{k-1}), \end{aligned} \tag{4}$$

which is also a general framework for DA problems. The parameter vector θ_k contains the unknown parameters in the evolution function $M(\cdot|\theta)$.

An important form of inference for HSSMs is the sequential Monte Carlo (SMC) method. SMC method, also known as particle filter (PF), is a class of DA methods, whose original version is first proposed by Gordon et al. (1993). Gilks and Berzuini (2001) introduce Markov chain Monte Carlo (MCMC) methods in PF, which can enhance particle diversity to alleviate the degeneracy problem and sample impoverishment in PF (Gustafsson 2010; Elfring et al. 2021). The framework combining MCMC with PF, also known as Particle Filter Markov Chain Monte Carlo (PMCMC), explores hidden states by PF, while parameters are estimated using an MCMC algorithm (Knape and De Valpine 2012). PF methods and MCMC methods mutually benefit each other in the problem of inference in HSSMs (Andrieu et al. 2010). Vrugt et al. proposed Particle-DREAM method which combines the strengths of sequential Monte Carlo sampling and Markov chain Monte Carlo simulation and is especially designed for the treatment of forcing, parameter, model structural, and calibration data error (Vrugt et al. 2013). Moradkhani et al. proposed an improved PF algorithm

for hydrologic prediction using MCMC moves to increase parameter diversity within the posterior distribution (Moradkhani et al. 2012).

Except for combining with PFs, MCMC methods can also combine with ensemble Kalman filter (EnKF) for Bayesian inference in HSSMs. Katzfuss et al. (2020) introduced a class of ensemble filtering and smoothing algorithms, namely eMCMC by replacing PF methods in PFMCMC with EnKF. In their methods, both the ensemble of parameters and the ensemble of states are considered to realize both the parameter estimation and state update.

The aforementioned methods are all based on the assumption that the evolution model is deterministic, in which the equations and the dimension of the parameters are fixed. These methods are not applicable to problems of identifying unknown processes in the evolution models, because the dimension of the model parameters may change when the type of processes changes. To overcome this gap, we propose a new framework named Reversible Jump Ensemble Filter (RJEnF). In the framework, we design a new Reversible Jump Markov Chain Monte Carlo (RJMCMC) method for the DA field, whose transition kernel is designed based on ensemble filters. RJMCMC is a type of Bayesian inference method used for static parameter estimation in complex models where the dimension of the parameters can vary. Compared with the standard MCMC methods where the dimension of the model parameters remains fixed, RJMCMC allows for changes in the dimensionality of the model itself. This is achieved by including moves that change the number of parameters, such as adding or deleting parameters. RJMCMC can be effectively applied to data assimilation problems, which allows RJMCMC to be used in dynamic systems with sequential acquired observational data. Wiese et al. (2015) combines PF with RJMCMC to estimate the directions of arrival of sources. Clay et al. (2021) utilizes the RJMCMC algorithm to select different Unscented Kalman Filters during the data assimilation process.

In this work, we proposed a new Bayesian DA framework that not only can update the states and model parameters, but also identify and update the model processes. This innovative framework is developed that uses data-driven methods for simultaneously and recursively identifying physical processes and estimating states and model parameters, which can get better predictions of states in the forecast step. This methodology, which combines the state and parameter update and evolution model identification together, has not been considered in the previous literature.

The rest of this article is structured as follows. In Sect. 2 we propose a DA method to perform Bayesian statistical inference on the states, model parameters, and the process of model. Three physical models are used to test and evaluate our method in Sect. 3, namely the linear spatio-temporal evolution model, the Lorenz 96 model, and the contaminant solute transport model. We discuss limitations, further extensions, and possible future research in Sect. 4.

2 Methodology

2.1 Sequential Monte Carlo algorithm

PF and EnKF are two important ensemble filters for Bayesian inference SSMs (4). Filtering for SSMs consists of two steps at every time point: a forecast step and an update step (Katzfuss et al. 2016, 2020). Assuming that the filtering distribution at the previous time k is given by

$$p(x_k|z_{1:k}), \tag{5}$$

where $z_{1:k} = \{z_1, z_2, \cdots, z_k\}$ denotes the observations available at time k. and the forecast step computes the forecast distribution at time k based on (2) as

$$p(x_k|z_{1:k-1}) = p(x_k|x_{k-1})p(x_{k-1}|z_{k-1}). \tag{6}$$

In this part, we assume that the modes in the equations within the evolution model $M(\cdot|\theta)$ from Eq. (4) is known, but $\theta_{1:k}$ are unknown. The Bayesian filtering problem requires computing the joint posterior distribution $p(x_k, \theta_k|z_{1:k})$ of the state and the parameters at each time $k = 1, 2, \cdots, T$. This joint posterior integrates all available information about the states and parameters contained in the prior and observations, and it is typically summarized through marginal distributions, posterior means, standard deviations, or credible intervals, which accounts for parameter uncertainty.

The joint posterior distribution is unavailable in closed form, so Monte Carlo methods must be used to approximate the distribution. As θ here is the parameter in evolution function $M(\cdot|\theta)$, it is logical to use the decomposition of the joint posterior distribution of the state and parameters into two terms: the conditional posterior distribution for the states given the parameters and the marginal posterior distribution for the parameters (Stroud et al. 2018):

$$p(x_k, \theta_k|z_{1:k}) = p(x_k|\theta_k, z_{1:k})p(\theta_k|z_{1:k}). \tag{7}$$

The problem we focus on is identifying the processes dynamically with sequentially acquired observations. Therefore, our framework is based on SMC. We have the initial information on the parameter and state in the modes in prior distribution $p_0(x)$ and $p_0(\theta)$.

Then we focused on the forecast step and update step at a single time point k, $k = 1, 2, \cdots$. Assume that there will not be drastic changes in the parameters at adjacent time steps, or multiple drastic changes in the parameters within a short period of time. At every time step, we can use the

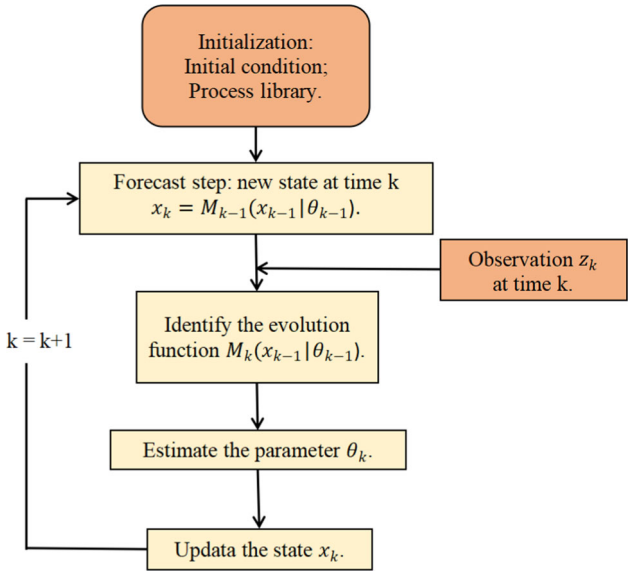


Fig. 1 The flow chart of the model-identification-fused DA

posterior distribution of the parameters $p(\theta_{k-1}|z_{k-1})$ from the previous time step as the prior distribution of the parameters $p_k(\theta_k)$ at the current time step. For the i-th ensemble member, we sample a proposal parameter $\theta^*_{i,k}$. Then we can run the evolution model with $\theta^*_{i,k}$:

$$x^*_{i,k} = M(\hat{x}^*_{i,k-1}|\theta^*_{i,k-1}) + \epsilon_{i,k-1}, \tag{8}$$

and the observation function

$$z^*_{i,k} = h(x^*_{i,k}) + v_{i,k}. \tag{9}$$

Then calculate the proposal weight

$$w^*_{i,k} = p_k(\theta^*_{i,k}) \cdot L(z^*_{i,k}|z_k), \tag{10}$$

where $L(z^*_{i,k}|z_k)$ is the likelihood function. Similarly, we can calculate the weight of the parameter $\theta_{i,k}$

$$w_{i,k} = p_k(\theta_{i,k}) \cdot L(z_{i,k}|z_k). \tag{11}$$

Then we can calculate the acceptance probability $w^*_{i,k}/w_{i,k}$ and choose whether to accept or reject the proposed parameters and get the new ensemble of parameters $\left\{\hat{\theta}^{(i)}_k\right\}^N_{i=1}$ as well as the new ensemble of states $\left\{\hat{x}^{(i)}_k\right\}^N_{i=1}$.

2.2 Reversible jump ensemble filter framework for dynamic model identification

The goal is to make sequential Bayesian inference on the potential evolution processes, states x_k and parameters θ_k for HSSMs (4). The core of the process identification lies in recognizing which $\phi_{j,k}(x, \alpha_{j,k})$ occurs in Eq. (1). Sometimes, data-driven methods are needed to identify the evolution model. Chang and Zhang (2019) proposed a method to identify the physical process via combined data-driven and data-assimilation methods and a set of fixed spatiotemporal measurement data, while this method is not applicable in DA framework. Usually, we cannot identify the candidate processes at the beginning when the observations are deficient. Moreover, the leading or governing equations may change as time goes by. Therefore, except for the reason that the observations are acquired sequentially, the modes in Eq. (1) cannot be determined at the beginning of the SMC algorithm, and can change during the process. So determining the evolution equation $M(\cdot|\theta)$ and inferring uncertain parameters θ of nonlinear models should be considered in a sequential way, which can be fused in the DA framework (See Fig. 1).

Figure 1 provides a schematic overview of our framework. At each time step, we first calculate the current state based on the previous state and parameter ensemble in the forecast step. When new observations are obtained, the update step performs identifying the process, estimating parameters, and updating the state.

Existing SMC methods are not suitable for the current situation because during process identification, the processes at adjacent time steps may change, resulting in the dimensionality changes of the parameters. Here, an ensemble of evolution function $\left\{M^{(i)}_k(\cdot|\cdot)\right\}^N_{i=1} \triangleq \left\{M^{(1)}_k(\cdot|\cdot), M^{(2)}_k(\cdot|\cdot), \ldots, M^{(N)}_k(\cdot|\cdot)\right\}$ is introduced to consider the uncertainty of the processes changes. Here we propose a sequential Monte Carlo method for Bayesian DA named Reversible Jump Ensemble Filter (RJEnF). Reversible Jump Markov Chain Monte Carlo (RJMCMC) is a Bayesian statistical method for the estimation of parameters in models where the number of parameters is unknown or variable. RJMCMC can be viewed as an extension of the Metropolis-Hastings algorithm more general state spaces, which was first proposed by Green (1995). RJMCMC is a type of MCMC algorithm that allows the posterior distribution to be explored over a space of models with different numbers of parameters. In this way, we can achieve sequential process identification, as well as parameter and state estimation.

For notational simplicity, we leave out the notation k for the time step here. In this study, we consider the number of the formats of the evolution equations to be finite. Suppose that for observed data z we have a countable collection of candidate evolution models $M = \{M_1, M_2, \cdots, M_{n_q}\}$ indexed by $q \in \mathbf{Q} \triangleq \{1, 2, \cdots, n_q\}$. Each model M_q has an r_q-dimensional vector of unknown parameters, $\theta_q \in \mathbb{R}^{r_q}$, where n_q can take different values for different models $q \in \mathbf{Q}$. The joint posterior distribution of (q, θ_q) is

$$\pi(q,\theta_q|z) = \frac{L\left(z|q,\theta_q\right)p(\theta_q|q)p(q)}{\sum_{q'\in\mathbf{Q}}\int_{\mathbb{R}^{n_{q'}}} L\left(z|q',\theta'_{q'}\right)p(\theta'_{q'}|q')p(q')d\theta'_{k'}}. \tag{12}$$

where $L(z|q,\theta_q)$ is the likelihood, and $p(q,\theta_q) = p(\theta_q|q)$ $p(q)$ is the joint prior distribution.

The reversible jump algorithm uses the joint posterior distribution in Eq. (12) as the target of a Markov chain Monte Carlo sampler over the parameter space where the dimension of (q,θ_q) can vary over the state space.

The difficulty in designing the reversible jump algorithm lies in determining the mapping functions and the transition kernel (Green and Hastie 2009; Fan and Sisson 2011). Let $p_{qq'}$ denote the probability of transition from q to q'. For the case of the finite set of potential models, the transition kernel can be expressed in the form of a transition matrix

$$T = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n_q} \\ q_{21} & q_{22} & \cdots & q_{2n_q} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n_q1} & q_{n_q2} & \cdots & q_{n_qn_q} \end{bmatrix}_{n_q\times n_q}. \tag{13}$$

Given the application scenario of ensemble DA, where there is an ensemble of evolution models $\left\{M_{k-1}^{(i)}(\cdot|\cdot)\right\}_{i=1}^{N}$. $M_{k-1}^{(i)}(\cdot|\cdot)$ represents an evolution model, which means every ensemble member of state corresponds to its own evolution model. The transition matrix T is designed based on the number of ensemble members representing different evolution models. Let N_q denote the number of ensemble members representing the evolution M_q, and $\sum_{i=1}^{n_q} N_q = N$. Let $p_{qq'} = N_{q'}/N$. In practice, we do not want the probability of remaining in the current state to be too high, as it hinders the exploration of other possibilities. At the same time, we do not want the probability of remaining in the current state to be too low, such that the evolution function followed by each particle changes frequently. Therefore, we set an upper bound q_{max} and a lower bound q_{min} on $p_{qq'}$ to ensure $q_{min} \le p_{qq'} \le q_{max}$. Finally, we proportionally scale the remaining transition probabilities to satisfy $\sum_{q=1}^{n_q} p_{qq'} = 1$. After defining the transition matrix, we can implement the RJEnF algorithm as Algorithm 1. We first determine a library $\Phi(x,\alpha)$ comprised of potential processes, and by selecting different potential processes from $\Phi(x,\alpha)$, we can construct various evolution models $M_k^{(i)}(\cdot|\cdot)$. The prior distribution $p_0(\theta)$ of the parameter is given to sample the ensemble of parameters. The algorithm starts with an ensemble of evolution functions $\left\{M_0^{(i)}(\cdot|\cdot)\right\}_{i=1}^{N}$, an ensemble of parameters $\left\{\theta_0^{(i)}\right\}_{i=1}^{N}$, and an ensemble of initial state $\left\{x_0^{(i)}\right\}_{i=1}^{N}$. For every time step k, The i-th ensemble member of $x_k^{(i)}$ is propagated by the evolution model $M_k^{(i)}(\cdot|\cdot)$ with $\theta_k^{(i)}$:

$$x_k^{(i)} = M_{k-1}^{(i)}(x_{k-1}^{(i)}|\theta_{k-1}^{(i)}). \tag{14}$$

The forecast state is given by

$$\tilde{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_k^{(i)} \tag{15}$$

When the observations z_k are obtained, the update step starts. For every ensemble member, a new evolution model $M_k^{*(i)}$ is proposed by transition matrix (13), then the proposal parameter $\theta^{*(i)}$ for $M_{k-1}^{*(i)}$ is generated. We can compare the likelihood functions to decide whether to accept or reject the proposed evolution model and parameter. After the acceptance/rejection step, the analyzed state $\hat{x}_k$ can be updated in different ways depending on which DA method is used. By the acceptance/rejection step, the ensemble members of parameters converge to the true parameter.

Algorithm 1 RJEnF Framework

Initialization: Library of latent processes $\Phi(x,\alpha)$;
1: The ensemble of evolution functions $\left\{M_0^{(i)}(\cdot|\cdot)\right\}_{i=1}^{N}$;
2: The ensemble of parameters $\left\{\theta_0^{(i)}\right\}_{i=1}^{N}$;
3: The ensemble of initial state $\left\{x_0^{(i)}\right\}_{i=1}^{N}$;
4: The prior distribution of parameter $p(\theta)$.
For time step k, $k = 1, 2, \cdots$: New observation z_k.
5: **for** $i = 1, 2, \cdots, N$ **do**
6: $x_k^{(i)} = M_{k-1}^{(i)}(x_{k-1}^{(i)}|\theta_{k-1}^{(i)})$.
7: Propose a proposal evolution function $M_{k-1}^{*(i)}$ by transition matrix (13) and a proposal parameter $\theta^{*(i)}$ from $p_{k-1}(\theta)$.
8: Calculate $L(z_k|M_{k-1}^{(i)},\theta_{k-1}^{(i)})$ and $L\left(z_k|M_{k-1}^{*(i)},\theta_{k-1}^{*(i)}\right)$, and decide the $M_k^{(i)}$ and $\theta_k^{(i)}$.
9: **end for**
10: Update the distribution of θ as $p_k(\theta)$.
11: Update the ensemble of state.
12: $k = k+1$.

By combining RJEnF with different DA methods, we obtain 2 methods, namely Reversible Jump Ensemble Kalman Filter (RJEnKF) and Reversible Jump Particle Filter (RJPF). RJEnKF uses EnKF to update the states which is easy to implement. While RJPF is more suitable for nonlinear and non-Gaussian problems as it uses PF to update the states without linear and Gaussian assumptions.

Designing the transfer matrix T and sampling the proposal evolution model only increase some negligible computational cost. Similar to the PFMCMC algorithm, the computational cost of the RJEnF algorithm primarily lies in the evolution of the state vector by Eq. (8). For the more

intractable or computationally expensive evolution model, a threshold for the likelihood function can be introduced. For every ensemble member, the likelihood $L(z_k|M_{k-1}^{(i)}, \theta_{k-1}^{(i)})$ at time $k-1$ is easy to compute. In the propagation from time $k-1$ to k, for those ensemble members whose likelihoods are below the threshold, sampling the proposal evolution functions and parameters and accepting/rejecting steps follow the Algorithm 1. For the remaining ensemble members, these steps are omitted, which can reduce the total number of times of computing the propagation of states.

2.2.1 Reversible jump ensemble Kalman filter

As mentioned earlier, we have two methods to implement state updates in the analysis step, and one of these methods is the EnKF method when the evolution function $M(x)$ and observation function $h(x)$ in Eqs. (2) and (3) can be linearized as the matrix $M \in \mathbb{R}^{n\times n}$ and $H \in \mathbb{R}^{m\times n}$ respectively, which is named as Reversible Jump Ensemble Kalman Filter (RJEnKF).

For EnKF, the forecast step computes the forecast distribution at time k based on (2) as

$$x_k|z_{1:k-1} \sim \mathcal{N}_n(\widetilde{x}_k, \widetilde{P}_k), \tag{16}$$

where $\widetilde{x}_k$ and $\widetilde{P}_k$ are the forecast mean and covariance matrix at time k. The filtering distribution at the previous time k is given by

$$x_k|z_{1:k} \sim \mathcal{N}_n(\hat{x}_k, \hat{P}_k), \tag{17}$$

where $\hat{x}_k$ and $\hat{P}_k$ are the updated mean and covariance matrix with new observation z_k. The update step modifies the forecast distribution using the new observations z_k at time k. Assume that the ensemble $\left\{x_k^{(i)}\right\}_{i=1}^N \triangleq \left\{x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(N)}\right\}$ is a set of N samples from the filtering distribution at time k. The EnKF then propagates each state vector forward and acquires the forecast state $x_k^{f,(i)}$ by

$$x_k^{f,(i)} = M_{k-1}(x_{k-1}^{(i)}) + \epsilon_{k-1}^{(i)}, i = 1, 2, \cdots, N, \tag{18}$$

and forecast mean $\widetilde{x}_k$ and the forecast covariance matrix are given by

$$\widetilde{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_k^{f,(i)}, \tag{19}$$

and

$$\widetilde{P}_k = C \circ \left[\frac{1}{N-1}\sum_{i=1}^{N}\left(x_k^{f,(i)} - \widetilde{x}_k\right)\left(x_k^{f,(i)} - \widetilde{x}_k\right)'\right], \tag{20}$$

which is the element-wise product of the empirical forecast covariance matrix with a sparse tapering correlation matrix C for implementing localization (Farchi and Bocquet 2018). As in most applications, we have $n \gg N$, and some form of regularization of this covariance matrix is necessary. The Kalman gain is calculated by

$$K_k = \widetilde{P}_k H'[H\widetilde{P}_k H' + R_k]. \tag{21}$$

At the update step, every ensemble member of states $x_k^{(i)}$ is updated by

$$x_k^{a,(i)} = x_k^{(i)} + K_k[z_k - Hx_k^{f,(i)} + v_k^{(i)}], \tag{22}$$

where $v_k^{(i)} \sim N(0, R_k)$. The mean and covariance matrix of states are updated with the new observation z_k by

$$\widehat{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_k^{a,(i)}, \tag{23}$$

and

$$\widehat{P}_k = C \circ \left[\frac{1}{N-1}\sum_{i=1}^{N}\left(x_k^{a,(i)} - \widehat{x}_k\right)\left(x_k^{a,(i)} - \widehat{x}_k\right)'\right]. \tag{24}$$

Besides the ensemble of initial state $\left\{x_0^{(i)}\right\}_{i=1}^N$, RJEnKF also requires the library of latent processes $\Phi(x, \alpha)$, the ensemble of evolution functions $\left\{M_0^{(i)}(\cdot|\cdot)\right\}_{i=1}^N$, and the ensemble of parameters $\left\{\theta_0^{(i)}\right\}_{i=1}^N$ as the initialization. We also need the assumption of the observation error matrix R_k, which is usually considered as a diagonal matrix.

The algorithm of the RJEnKF is described in Algorithm 2. In every analyse step at time k, and for i-th ensemble member, the evolution model is

$$x_k^{f,(i)} = M_{k-1}^{(i)}(x_{k-1}^{(i)}|\theta_{k-1}^{(i)}) + \epsilon_{k-1}^{(i)}, i = 1, 2, \cdots, N. \tag{25}$$

To realize the identification of the evolution model, proposal evolution model $M_{k-1}^{*(i)}$ and a proposal parameter $\theta^{*(i)}$ are sampled. After the acceptance/rejection step, a more appropriate pair of evolution model and parameter is kept for the i-th ensemble member. Then the forecast state and its forecast covariance matrix are given by Eqs. (19) and (20).

2.2.2 Reversible jump particle filter

When the evolution model is strongly nonlinear, the update of states should be calculated using PF. Instead of the Gaussian assumption, the PF approximates the probability density function (pdf) representing the posterior by a discrete pdf

Algorithm 2 RJEnKF

Initialization: Library of latent processes $\Phi(x, \alpha)$;
The ensemble of evolution functions $\left\{M_0^{(i)}(\cdot|\cdot)\right\}_{i=1}^N$;
The ensemble of parameters $\left\{\theta_0^{(i)}\right\}_{i=1}^N$;
The ensemble of initial state $\left\{x_0^{(i)}\right\}_{i=1}^N$;
The prior distribution of parameter $p(\theta)$;
The observation error matrix R_k
For time step k, $k = 1, 2, \cdots$: New observation z_k.
for $i = 1, 2, \cdots, N$ **do**
  $x_k^{(i)} = M_{k-1}^{(i)}(x_{k-1}^{(i)}|\theta_{k-1}^{(i)})$.
  Propose a proposal evolution function $M_{k-1}^{*(i)}$ by transition matrix (13) and a proposal parameter $\theta^{*(i)}$ from $p_{k-1}(\theta)$.
  Calculate $L(z_k|M_{k-1}^{(i)}, \theta_{k-1}^{(i)})$ and $L\left(z_k|M_{k-1}^{*(i)}, \theta_{k-1}^{*(i)}\right)$, and decide the $M_k^{(i)}$ and $\theta_k^{(i)}$.
  Propagate the ensemble member $x_k^{(i)}$ through the corresponding $M_k^{(i)}$ and $\theta_k^{(i)}$ decided in the previous step by $x_k^{(i)} = M_k^{(i)}(x_{k-1}^{(i)}|\theta_k^{(i)})$
end for
Update the distribution of θ as $p_k(\theta)$.
Calculate the forecast state and its covariance matrix by Eqs. (19) and (20).
Calculate the Kalman gain by Eq. (21).
Update all the ensemble members of states by Eq. (22).
$k = k + 1$.

such that the distributions of states and parameters do not have to fit the Gaussian assumption. The distribution is approximated by a sum of weighted samples:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N} \omega_k^{(i)} \delta(x_k - x_k^{(i)}). \tag{26}$$

If we consider an ensemble of parameter θ as $\left\{\theta_k^{(i)}\right\}_{i=1}^N \triangleq \left\{\theta_k^{(1)}, \theta_k^{(2)}, \ldots, \theta_k^{(N)}\right\}$, then we have

$$p(x_k, \theta_k|z_{1:k}) \approx \sum_{i=1}^{N} \omega_k^{(i)} \delta((x_k, \theta_k) - (x_k^{(i)}, \theta_k^{(i)})). \tag{27}$$

Here $\left\{\omega_k^{(i)}, x_k^{(i)}, \theta_k^{(i)}\right\}_{i=1}^N$ is an ensemble containing N members of state and parameter and weights. Each member $\{x_k^{(i)}, \theta_k^{(i)}\}$ represents a possible realization of the state and parameter sequence. A weight $\omega_k^{(i)}$ represents the relative importance of each of the N samples and $\sum_{i=1}^N \omega_k^{(i)} = 1$. Samples associated with high weights are believed to be closer to the true state sequence than samples associated with low weights. $\delta(\cdot)$ denotes the Dirac delta function:

$$\delta(x_k, \theta_k) = \begin{cases} 1, & \text{if } (x_k, \theta_k) = (x_k^{(i)}, \theta_k^{(i)}), \\ 0, & \text{if } (x_k, \theta_k) \neq (x_k^{(i)}, \theta_k^{(i)}). \end{cases} \tag{28}$$

The weights $\omega_k^{(i)}$ are given by

$$\omega_k^{(i)} = \frac{p(z_k|x_k^{(i)}, \theta_k^{(i)})}{\sum_{j=1}^N p(z_k|x_k^{(j)}, \theta_k^{(j)})}. \tag{29}$$

For each time k, in the update step, the weights of the members are calculated by Eq. (29) based on the observation z_k, and the posterior distribution is given by Eq. (27). In the forecast step, each member is propagated by the evolution function, and the joint forecast distribution is

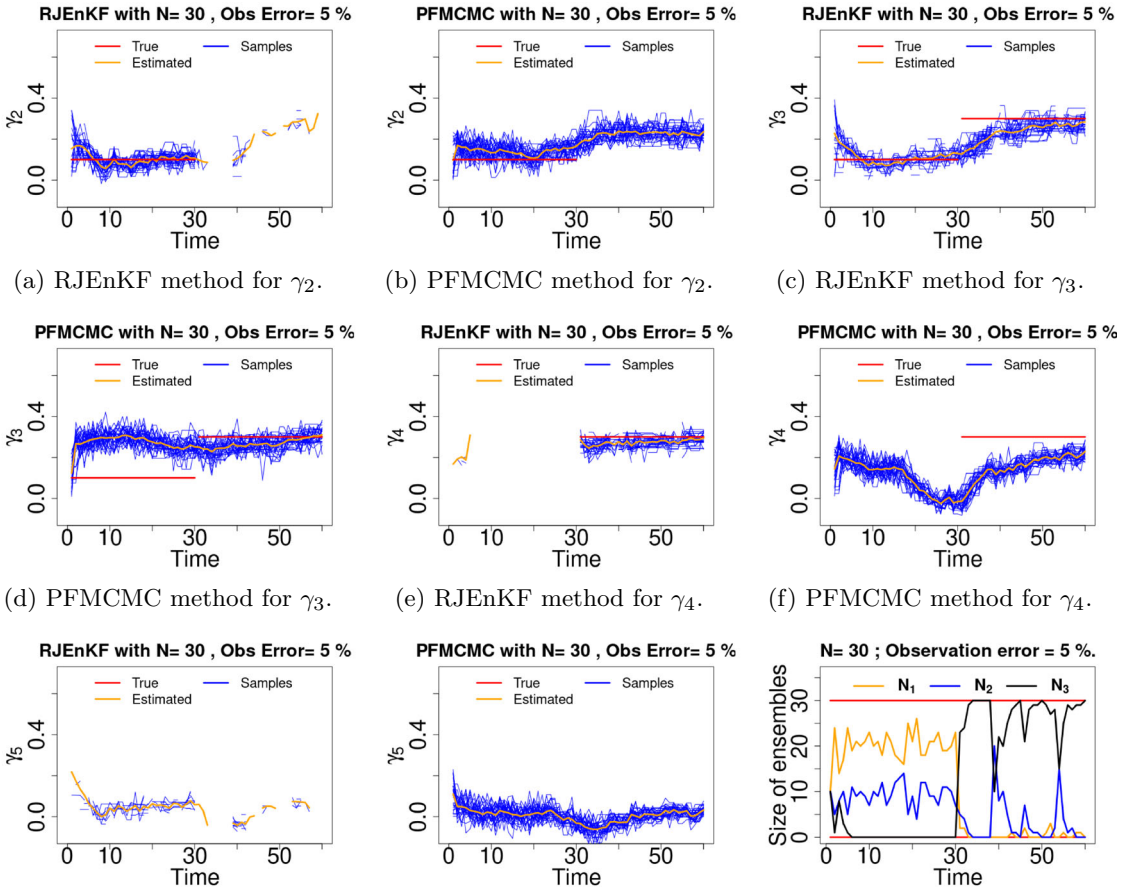
$$p^f(x_k, \theta_k|z_{1:k-1}) \approx \sum_{i=1}^{N} \omega_{k-1}^{(i)} \delta((x_k, \theta_k) - (x_k^{(i)}, \theta_k^{(i)})). \tag{30}$$

If the process of propagation of the ensemble and assimilation of new observations are repeated a few times (or with a large number of observations only once), only one member with a large weight will remain and all others have negligible weight. This is called particle collapse or degeneration, which means that the statistical information in the ensemble is lost; effectively only one particle has all information available to us. A way to avoid this is the so-called resampling. Vrugt et al. (2013) and Moradkhani et al. (2012) proposed resampling methods based on MCMC methods, which can relieve the collapse of PF.

Based on the concept of sequential importance sampling and the use of Bayesian theory, PF is particularly useful in dealing with nonlinear and non-Gaussian problems. In PF, the distributions are approximated by discrete random measures defined by particles and weights $w_k^{(i)}$ assigned to the particles:

$$\mathcal{E} = \{M_k^{(i)}(\cdot|\cdot), \theta_k^{(i)}, x_k^{(i)}, w_k^{(i)}\}_{i=1}^N. \tag{31}$$

If the total uncertainty in the system becomes too large, there will be too few samples with meaningful weights, leading to the collapse of the ensemble, or weight degeneration (Snyder et al. 2008). There are many different PF methods, some of which avoid particle degeneracy by using resampling algorithms. Here we choose a simple resampling scheme which is suggested by van Leeuwen et al. (2019). The threshold of resampling is determined by drawing a sample value from the uniform distribution $U[0, 1/N]$. Then the ensemble members whose weights are under the threshold are replaced by the resampling members. The Pseudo-code of RJPF is given in Algorithm 3. To illustrate the effectiveness of RJPF, we use the Lorenz 96 model as a nonlinear problem in Sect. 3.2.



(a) RJEnKF method for γ_2. (b) PFMCMC method for γ_2. (c) RJEnKF method for γ_3.

(d) PFMCMC method for γ_3. (e) RJEnKF method for γ_4. (f) PFMCMC method for γ_4.

(g) RJEnKF method for γ_5. (h) PFMCMC method for γ_5. (i) Size of the ensemble members.

Fig. 2 The estimation of the parameters and the identification of models in Sect. 3.1 with $N = 30$ and 5% observation error. The sub-figures (**a**–**h**) show the estimation of the parameters by RJEnKF and PFMCMC. The red lines represent the true parameters; the blue lines represent the predicted parameters for each particle, and the orange lines represent the estimated parameter. From sub-figure (**a**), it can be seen that when the evolution model in the second stage does not include the parameter γ_2, only a few particles choose the evolution model containing γ_2 at certain time points. The sub-figure (i) shows the identification of models by the different sizes of the ensemble members representing different models changing with time. The different colors represent different models

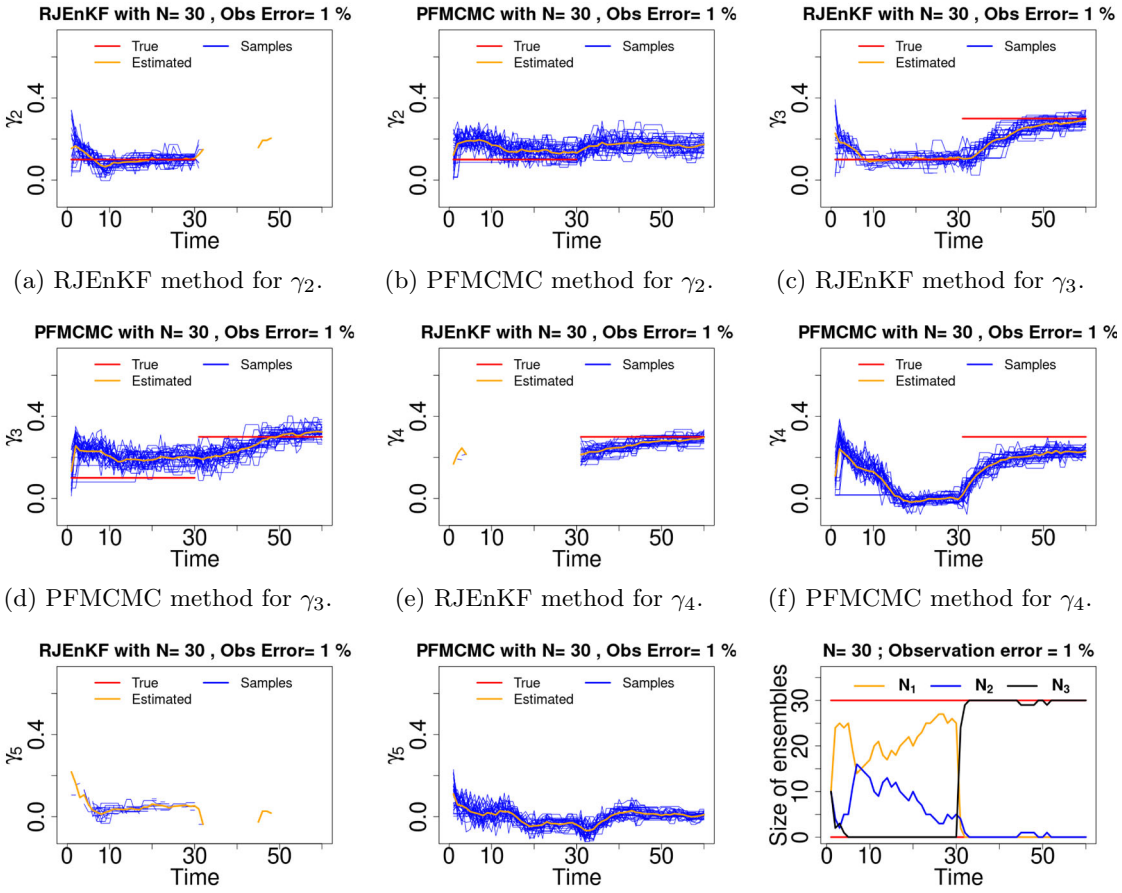
3 Case study

3.1 Linear evolution

First, we evaluate the proposed RJEnKF method by considering the linear dynamic spatiotemporal model (Xu and Wikle 2007; Stroud et al. 2018), but we pay attention to the parameters in the evolution model instead of the parameters in the observation function and covariance matrices. Our goal is to identify the true model, estimate the model parameters, and assimilate the states step-wisely. The model is a vector autoregression plus noise, where the state vector $x_k = (x_{k_1}, \cdots, x_{k_n})'$ corresponds to n equally spaced locations $\{1, 2, 3, \cdots, n\}$ along a spatial transect. Following the notation in (2) and (3), the evolution function is linear, $\mathcal{M}(x_{k-1}) = Mx_{k-1}$ where the propagator matrix is pentadiagonal with parameters $\gamma = (\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5)'$:

$$M = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_4 & & & 0 \\ \gamma_3 & \gamma_1 & \gamma_2 & \ddots & & \\ \gamma_5 & \gamma_3 & \gamma_1 & \ddots & \ddots & \\ & \ddots & \ddots & \gamma_1 & \gamma_2 & \gamma_4 \\ & & \ddots & \gamma_3 & \gamma_1 & \gamma_2 \\ 0 & & & \gamma_5 & \gamma_3 & \gamma_1 \end{bmatrix} \tag{32}$$

We consider the situation where γ_2 and γ_4 have exactly one zero, and γ_3 and γ_5 have exactly one zero. Then there are 3 potential processes, and the corresponding evolution



(a) RJEnKF method for γ_2. (b) PFMCMC method for γ_2. (c) RJEnKF method for γ_3.

(d) PFMCMC method for γ_3. (e) RJEnKF method for γ_4. (f) PFMCMC method for γ_4.

(g) RJEnKF method for γ_5. (h) PFMCMC method for γ_5. (i) Size of the ensemble members.

Fig. 3 The estimation of the parameters and the identification of models in Sect. 3.1 with $N = 30$ and 1% observation error

Algorithm 3 RJPF

Initialization: Library of latent processes $\Phi(x, \alpha)$;

1: The ensemble of evolution functions $\left\{M_0^{(i)}(\cdot|\cdot)\right\}_{i=1}^{N}$;

2: The ensemble of parameters $\left\{\theta_0^{(i)}\right\}_{i=1}^{N}$;

3: The ensemble of initial state $\left\{x_0^{(i)}\right\}_{i=1}^{N}$;

4: The initial weights $\omega_0^{(i)} = 1/N$ The prior distribution of parameter $p(\theta)$.

For time step k, $k = 1, 2, \cdots$**:** New observation z_k.

5: **for** $i = 1, 2, \cdots, N$ **do**

6: $x_k^{(i)} = M_{k-1}^{(i)}(x_{k-1}^{(i)}|\theta_{k-1}^{(i)})$.

7: Calculate the weights $\omega^{(i)}$ and normalize the weights.

8: **end for**

9: **for** $\omega^{(i)} > u \sim U[0, 1/n]$ **do**

10: Propose a proposal evolution function $M_{k-1}^{*(i)}$ by transition matrix (13) and a proposal parameter $\theta^{*(i)}$ from $p_{k-1}(\theta)$.

11: Calculate $L(z_k|M_{k-1}^{(i)}, \theta_{k-1}^{(i)})$ and $L\left(z_k|M_{k-1}^{*(i)}, \theta_{k-1}^{*(i)}\right)$, and decide the $M_k^{(i)}$ and $\theta_k^{(i)}$.

12: **end for**

13: Recalculate the weights $\omega^{(i)}$.

14: Update the distribution of θ as $p_k(\theta)$.

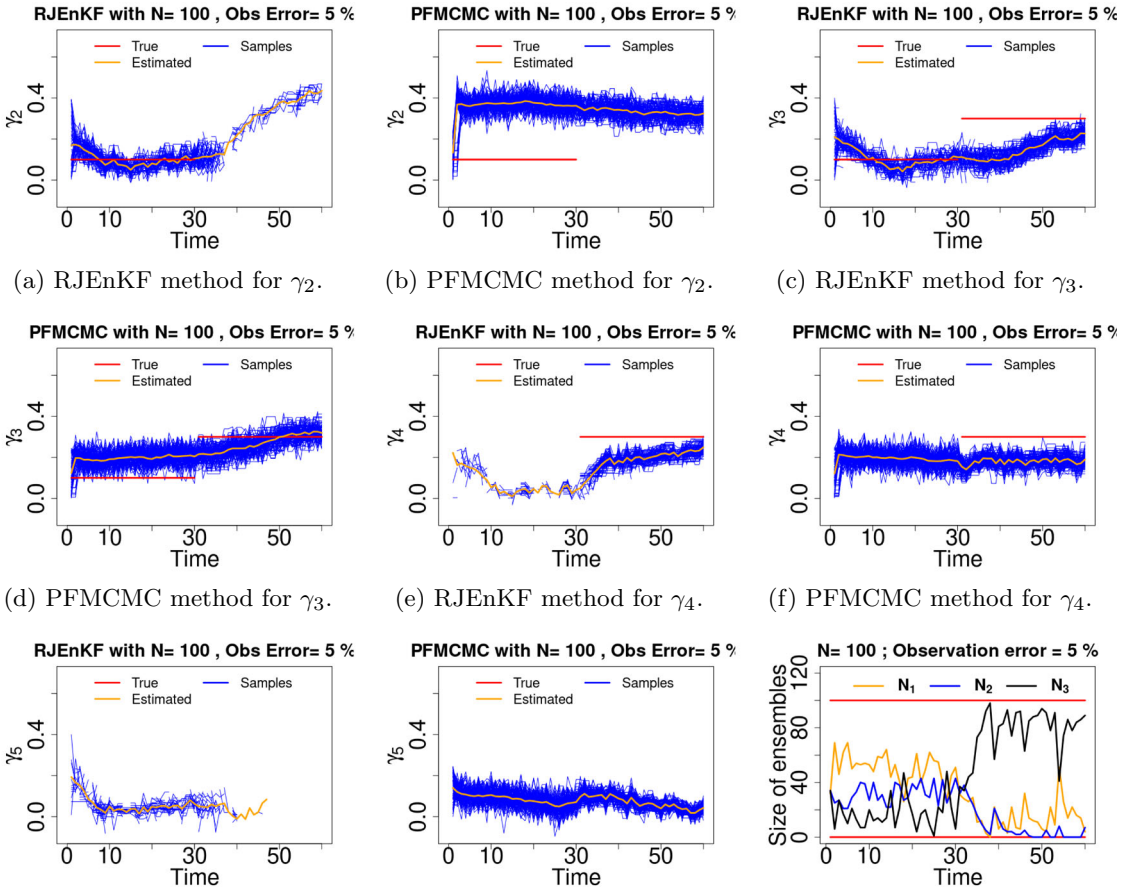
15: Update the ensemble of state.

16: $k = k + 1$.

matrices for Mode 1, Mode 2, and Mode 3 are denoted as M_1, M_2, and M_3, respectively:

$$M_1 = \begin{bmatrix} \gamma_1 & \gamma_2 & 0 & & 0 \\ \gamma_3 & \gamma_1 & \gamma_2 & \ddots & \\ 0 & \gamma_3 & \gamma_1 & \ddots & \ddots \\ & \ddots & \ddots & \gamma_1 & \gamma_2 & 0 \\ & & \ddots & \gamma_3 & \gamma_1 & \gamma_2 \\ 0 & & & 0 & \gamma_3 & \gamma_1 \end{bmatrix},$$

$$M_2 = \begin{bmatrix} \gamma_1 & \gamma_2 & 0 & & 0 \\ 0 & \gamma_1 & \gamma_2 & \ddots & \\ \gamma_5 & 0 & \gamma_1 & \ddots & \ddots \\ & \ddots & \ddots & \gamma_1 & \gamma_2 & 0 \\ & & \ddots & 0 & \gamma_1 & \gamma_2 \\ 0 & & & \gamma_5 & 0 & \gamma_1 \end{bmatrix},$$



(a) RJEnKF method for γ_2. (b) PFMCMC method for γ_2. (c) RJEnKF method for γ_3.

(d) PFMCMC method for γ_3. (e) RJEnKF method for γ_4. (f) PFMCMC method for γ_4.

(g) RJEnKF method for γ_5. (h) PFMCMC method for γ_5. (i) Size of the ensemble members.

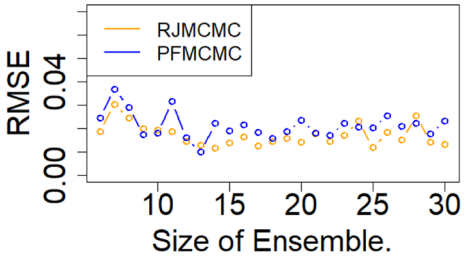
Fig. 4 The estimation of the parameters and the identification of models in Sect. 3.1 with $N = 100$ and 5% observation error

$$M_3 = \begin{bmatrix} \gamma_1 & 0 & \gamma_4 & & 0 \\ \gamma_3 & \gamma_1 & 0 & \ddots & \\ 0 & \gamma_3 & \gamma_1 & \ddots & \ddots \\ & \ddots & \ddots & \gamma_1 & 0 & \gamma_4 \\ & & \ddots & \gamma_3 & \gamma_1 & 0 \\ 0 & & & 0 & \gamma_3 & \gamma_1 \end{bmatrix}.$$

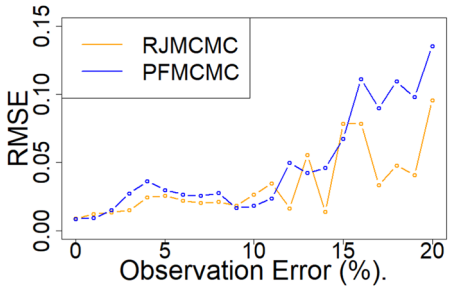
We denote the parameters of 3 different models as $\theta_1 = (\gamma_1, \gamma_2, \gamma_3)' \in \Theta_1$, $\theta_2 = (\gamma_1, \gamma_2, \gamma_5)' \in \Theta_2$, and $\theta_3 = (\gamma_1, \gamma_3, \gamma_4)' \in \Theta_3$ respectively. For all 3 models, the sum of the parameters should be 1. For example, if the evolution process obeys model 1, then $\gamma_1 + \gamma_2 + \gamma_3 = 1$. First, we simulated the true states $x_{true,k}, k = 1, 2, \cdots, m$, from the true model with dimensions $n = m = 20$ for $T = 60$ time points, with the first stage $t \in [1, 2, \cdots, 30]$, and the second stage $t \in [31, 32, \cdots, 60]$. The true evolution model is taken to be M_1 with true parameter $\theta_{true,1} = (0.8, 0.1, 0.1)'$ for the first stage, and M_3 with true parameter $\theta_{true,3} = (0.4, 0.3, 0.3)'$ for the second stage. Then we synthetically add noise to the true data as:

$$z_k = x_{true,k} \times (1 + \delta \times e), \tag{33}$$

where δ denotes the level of observation error; and e denotes the uniform random variable taking values from -1 to 1. Considering $\delta = 5\%$ here, we compare our method with the classical PFMCMC method. For the PFMCMC method, since it is unable to choose between different models, we need to consider the evolution matrix M in Eq. (32), and the 5-dimensional parameter $\theta \triangleq (\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5)' \subset \Theta$, with $\sum_{i=1}^{5} \gamma_i = 1$. For all the models, γ_1 is determined by other parameters, so for M_1, M_2, and M_3, the numbers of unknown parameters are 2; while for M, the number of unknown parameters is 4. For both methods, we choose the number of ensemble members N as 30. The results are shown in Fig. 2. In Fig. 2a–h, the red lines represent the true parameters. The blue lines are the ensemble members, and the orange lines represent the estimation of parameters. For γ_3, the results of RJEnKF can converge to the true parameters in both stages (see Fig. 2c), while the results of PFMCMC could not converge to the true parameter in the first stage (see Fig. 2d). For γ_4, in Fig. 2e, the estimation of the parameter

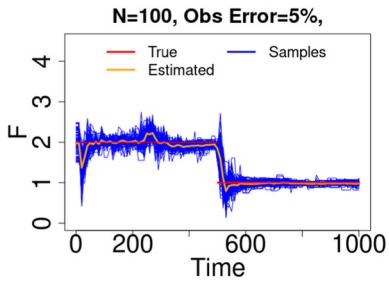


(a) RMSE versus size of ensemble.

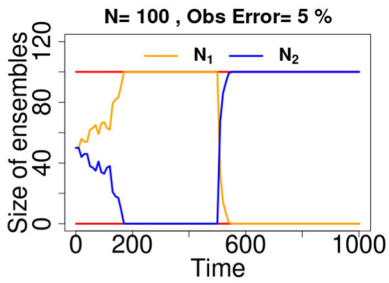


(b) RMSE versus observation error.

Fig. 5 The root mean squared error(RMSE) of prediction for RJEnKF and PFMCMC in Sect. 3.1



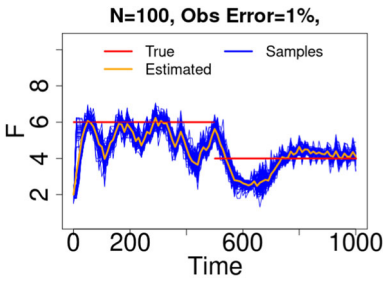
(a) The prediction of parameter F.



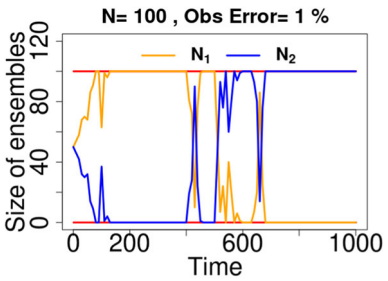
(b) Size of the ensemble members.

Fig. 6 The estimation of the parameters and the identification of models in Sect. 3.2 for RJPF

through RJEnKF method converges to the true parameter, while the PFMCMC could not converge to the true parameter, as shown in Fig. 2f. Figure 2i shows the sizes of ensemble members representing the different models. In the first stage, the number of the ensemble members representing model 1 is the largest, which identifies the model 1 as the true model. As



(a) The prediction of parameter F.



(b) Size of the ensemble members.

Fig. 7 The estimation of the parameters and the identification of models in Sect. 3.2 for RJPF when the dynamic system is more chaotic

long as it comes to stage 2, the number of the ensembles for model 3 dominates. When the dynamic changes at time point 31, the number of ensemble members representing evolution model 1 suddenly decreases, while the number of ensemble members representing evolution model 3 suddenly increases. It shows that the RJEnKF method is sensitive to the change of the evolution model, which means that RJEnKF can identify the processes in the dynamic system as well as their changes.

To study the method's sensitivity to the observation error, let $\delta = 1\%$. Figure 3 shows the results, in which the identification of the models for both two stages gets better. Especially for the second stage, almost all of the ensemble members are related to model 3, which means we are certain the true model is model 3. We also increase the size of the ensemble members to let $N = 100$. The results are shown in Fig. 4. The PFMCMC method still cannot provide better results. As for PFMCMC method, the evolution model is imprecise, and the dimension of parameters is higher. So the ensembles may converge to wrong results.

To demonstrate the effectiveness of the proposed method, we investigated the prediction errors of two methods, which are RJEnKF and PFMCMC (Moradkhani et al. 2012), for different sizes of ensembles and different observation errors. From Fig. 5a and b, one can see that the prediction errors by both two methods decrease when the ensemble sizes increase, and increase with the observation error. The prediction errors of RJEnKF are smaller than those of PFMCMC.

3.2 Nonlinear evolution case: Lorenz 96 model

We consider RJPF applied to the extensively used medium-dimensional dynamical system Lorenz'96 model (L96) (Lorenz 1996). This model, which is developed by Edward Lorenz, represents a nonlinear chaotic dynamic system. The evolution model is

$$\frac{dx_j}{dt} = -x_{j-1}(x_{j-2} - x_{j+1}) - x_j + F. \tag{34}$$

We define a different evolution equation that corresponds to the reverse propagation of the state in the Lorenz 96 equation:

$$\frac{dx_j}{dt} = -x_{j+1}(x_{j+2} - x_{j-1}) - x_j + F, \tag{35}$$

where $x = \{x_j;\ j = 1, ..., n\}$ is the $n-$dimensional state vector. The true solution of $(x_{1k}, x_{2k}, \cdots, x_{nk})^T$ with $\Delta t = 0.03$ and $n = 40$ is computed for $t \in [0, 30]$ from the initial condition $x_0 = (x_{1,0}, x_{2,0}, \cdots, x_{n,0})^T = (1 + 0.01, 1, \cdots, 1)^T$, which means that there are 1001 time steps in total.

For the first 501 time steps, the evolution model follows Eq. (34) with $F = 2$; then for the next 500 time steps, the evolution model follows Eq. (35) with $F = 1$. Synthetic observations are subsequently created by perturbing each data point of the reference solution with a uniformly distributed measurement error. We choose the observation error to be 5%, and the RJPF works well. From Fig. 6a, one can see that, for both stages, the ensemble of particles can accurately predict the true parameter F and identify the change of the evolution process. At the same time, as can be seen from Fig. 6b, it is evident that by the change of the size of the ensemble representing different processes, our method can dynamically identify the evolution processes. For example, in the second stage, the size of the ensemble representing model 2 is 100, which confidently identifies that the evolution process follows the true model 2.

When the model becomes more chaotic by letting $F = 6$ for the first stage, and $F = 4$ for the second stage (Albarakati et al. 2022; Vrugt et al. 2013), and let first 501 time steps follow Eq. (34), and the last 500 time steps follow Eq. (35). The estimation of the parameter F gets worse, even though the observation error is only 1%. Figure 7a presents the results of the predicted parameter F. From Fig. 7b, one can see that through the change of the ensemble size, the evolution process can still be identified.

3.3 Contaminant solute transport

Next, let's consider the example of contaminant solute transport mentioned at the beginning of the article. The library of latent process $\Phi = [\frac{\partial C}{\partial x}, \frac{\partial^2 C}{\partial x^2}, C^{a-1}\frac{\partial C}{\partial t}, \frac{1}{(1+KC)^2}\frac{\partial C}{\partial t}]$, where $C^{a-1}\frac{\partial C}{\partial t}$ and $\frac{1}{(1+KC)^2}\frac{\partial C}{\partial t}$ never appear together. The evolution function can be written as:

$$\frac{\partial C}{\partial t} = -\beta_1\frac{\partial C}{\partial x} + \beta_2\frac{\partial^2 C}{\partial x^2} - f(C, \alpha_3)\frac{\partial C}{\partial t}, \tag{36}$$

where

$$f\,(c, \alpha_3) = \begin{cases} 0, & No - SORP, \\ C^{a-1}, & F - SORP, \\ \dfrac{1}{(1+KC)^2}, & L - SORP. \end{cases} \tag{37}$$

C is the concentration of solute in aqueous phase, and $\beta = [\beta_1, \beta_2, \beta_3]^T$ and $\alpha_3 = [a, K]^T$.

Three possible models corresponding to the three different sorption modes are considered here:

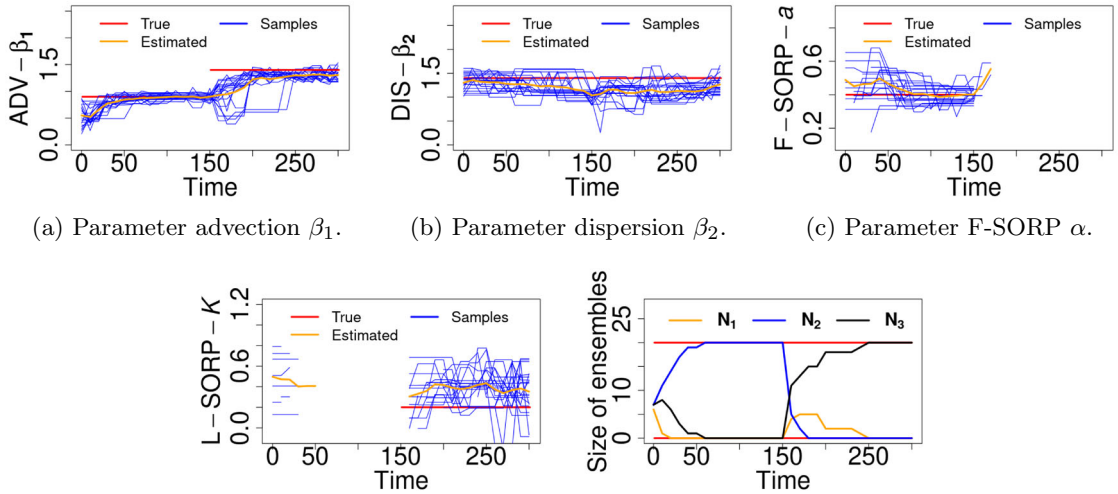
$$\begin{aligned} Model\ \ 1: &\ \frac{\partial C}{\partial t} = -\beta_1\frac{\partial C}{\partial x} + \beta_2\frac{\partial^2 C}{\partial x^2} \\ Model\ \ 2: &\ \frac{\partial C}{\partial t} = -\beta_1\frac{\partial C}{\partial x} + \beta_2\frac{\partial^2 C}{\partial x^2} - C^{a-1}\frac{\partial C}{\partial t} \\ Model\ \ 3: &\ \frac{\partial C}{\partial t} = -\beta_1\frac{\partial C}{\partial x} + \beta_2\frac{\partial^2 C}{\partial x^2} - \frac{1}{(1+KC)^2}\frac{\partial C}{\partial t} \end{aligned} \tag{38}$$

The aim is to obtain a full probabilistic description of the posterior probabilities of potential models, the parameters and the states with the sequential observed data. The parameter is $\theta = (\beta_1, \beta_2, a, K)^T$.

We consider 300 time steps, and the true evolution function follows Model 2 for the first 150 time steps and Model 3 for the following 150 time steps. In the first stage, which contains the first 150 time steps, the true parameter is $\theta_{true,stage1} = (\beta_1, \beta_2, a) = (0.9, 1.4, 0.4)^T$. In the second stage, the true parameter is $\theta_{true,stage2} = (\beta_1, \beta_2, K) = (1.4, 1.4, 0.2)^T$. In this case, the prior distribution $p(\theta)$ of the parameters $(\beta_1, \beta_2, a, K)^T$ is assumed to be a normal distribution $N(\mu_0, P_0)$, with $\mu_0 = (0.4, 1.3, 0.3, 0.3)^T$, and $P_0 = diag(0.2, 0.2, 0.1, 0.2)$. We set the size of the ensemble to be $N = 20$. The observations are acquired every 20 time steps with 5% observation errors.

As can be seen from Fig. 8, the RJEnKF method is able to identify the potential evolution equations which is shown through the sizes of ensembles corresponding to each model in Fig. 8e.

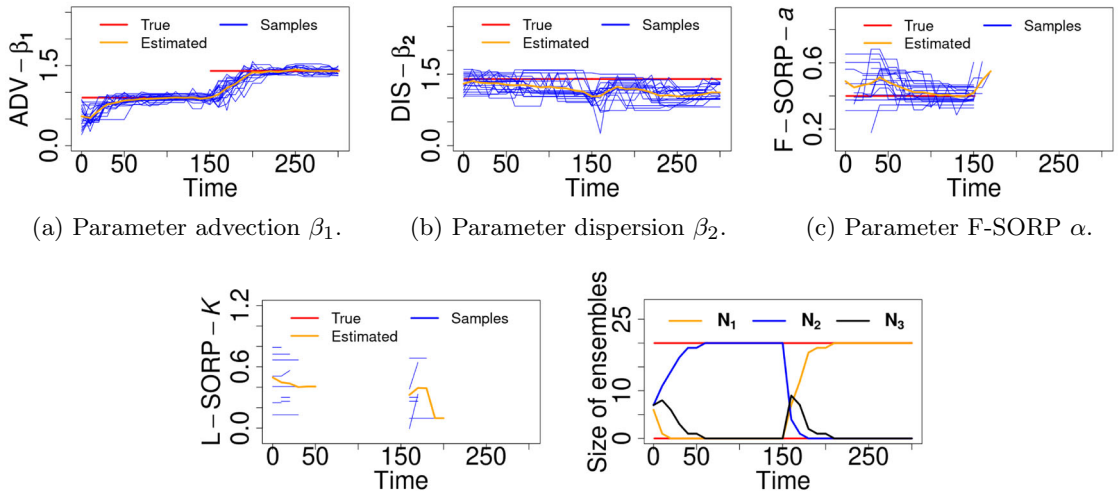
In the second experiment, we keep the same setting of the previous experiment, which is model 2 and $\theta_{true,stage1} = (\beta_1, \beta_2, a) = (0.9, 1.4, 0.4)^T$; the second stage follow the model 1 with parameter $\theta_{true,stage2} = (\beta_1, \beta_2) =$



(a) Parameter advection β_1. (b) Parameter dispersion β_2. (c) Parameter F-SORP α.

(d) Parameter L-SORP K. (e) Size of the ensembles.

Fig. 8 The identification of the models and estimation of the parameters for the example of contaminant solute transport. The red lines represent the true parameters; the blue lines represent the predicted parameters for each particle, and the orange lines represent the estimated parameter



(a) Parameter advection β_1. (b) Parameter dispersion β_2. (c) Parameter F-SORP α.

(d) Parameter L-SORP K. (e) Size of the ensemble members.

Fig. 9 The identification of the models and estimation of the parameters for the example of contaminant solute transport. The red lines represent the true parameters; the blue lines represent the predicted parameters for each particle, and the orange lines represent the estimated parameter

$(0.9, 1.4)^T$. The results are shown in Figure. 9, in which the true parameters can be estimated well again, and the true processes (see Fig. 9a–d) are correctly identified (see Fig. 9e).

4 Discussion and conclusion

The ability to dynamically and correctly identifying the underlying modes of the evolution model can improve the accuracy of DA methods. We have introduced a new class of DA methods which can solve three tasks together: determining the evolution equation, which includes identifying the occurring (or dominant) processes and selecting the proper empirical models; estimating the uncertain model parameters in the evolution equation; and updating the states based on the sequential observations.

The RJEnF framework we propose here is very flexible. On the one hand, the RJEnF framework can be combined with different DA methods to adapt to various problems, namely RJEnKF and RJPF. On the other hand, if we leave out the 'jump' step, the method reduces to the fixed dimension SMC (Vrugt et al. 2013; Moradkhani et al. 2012; Katzfuss et al. 2020). The algorithm introduces RJMCMC from statistics literature to the DA community, which can identify the occur-

ring processes fused in the DA methods. The main advantage of our algorithm is that it can solve the DA problem with undecided modes in evolution models, that has not been discussed in the DA literature before. We test and evaluate the proposed methods with both linear and nonlinear examples.

Author Contributions Y.H. and H.X.L conceived the research study. Y.H. contacted the experiments, analyzed the results and wrote the manuscript. H.X.L analyzed the results and reviewed the manuscript.

Funding Y.H. is supported by the China Scholarship Council.

Declarations

Conflict of interest The authors declare no conflict of interest.

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

References

Albarakati, A., Budišić, M., Crocker, R., Glass-Klaiber, J., Iams, S., Maclean, J., Marshall, N., Roberts, C., Van Vleck, E.S.: Model and data reduction for data assimilation: particle filters employing projected forecasts and data with application to a shallow water model. Comput. Math. Appl. **116**, 194–211 (2022)

Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **72**(3), 269–342 (2010)

Bachmann, K., Keil, C., Craig, G.C., Weissmann, M., Welzbacher, C.A.: Predictability of deep convection in idealized and operational forecasts: effects of radar data assimilation, orography, and synoptic weather regime. Mon. Weather Rev. **148**(1), 63–81 (2020)

Barthélémy, S., Brajard, J., Bertino, L., Counillon, F.: Super-resolution data assimilation. Ocean Dyn. **72**(8), 661–678 (2022)

Chang, H., Zhang, D.: Identification of physical processes via combined data-driven and data-assimilation methods. J. Comput. Phys. **393**, 337–350 (2019)

Clay, R., Ward, J.A., Ternes, P., Kieu, L.-M., Malleson, N.: Real-time agent-based crowd simulation with the reversible jump unscented Kalman filter. Simul. Model. Pract. Theory **113**, 102386 (2021)

Drovandi, C., Everitt, R.G., Golightly, A., Prangle, D.: Ensemble MCMC: accelerating pseudo-marginal MCMC for state space models using the ensemble Kalman filter. Bayesian Anal. **17**(1), 223–260 (2022)

Elfring, J., Torta, E., van de Molengraft, R.: Particle filters: a hands-on tutorial. Sensors **21**(2), 438 (2021)

Fan, Y., Sisson, S.A.: Reversible jump MCMC. In: Handbook of Markov Chain Monte Carlo, 67–92 (2011)

Farchi, A., Bocquet, M.: Comparison of local particle filters and new implementations. Nonlinear Process. Geophys. **25**(4), 765–807 (2018)

Gilks, W.R., Berzuini, C.: Following a moving target-Monte Carlo inference for dynamic Bayesian models. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **63**(1), 127–146 (2001)

Gordon, N.J., Salmond, D.J., Smith, A.F.: Novel approach to nonlinear/non-gaussian Bayesian state estimation. In: IEE proceedings F (radar and signal processing), **140**, 107–113. IET (1993)

Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika **82**(4), 711–732 (1995)

Green, P.J., Hastie, D.I.: Reversible jump MCMC. Genetics **155**(3), 1391–1403 (2009)

Gustafsson, F.: Particle filter theory and practice with positioning applications. IEEE Aerosp. Electron. Syst. Mag. **25**(7), 53–82 (2010)

Jin, J., Lin, H.X., Segers, A., Xie, Y., Heemink, A.: Machine learning for observation bias correction with application to dust storm data assimilation. Atmos. Chem. Phys. **19**(15), 10009–10026 (2019)

Katzfuss, M., Stroud, J.R., Wikle, C.K.: Understanding the ensemble Kalman filter. Am. Stat. **70**(4), 350–357 (2016)

Katzfuss, M., Stroud, J.R., Wikle, C.K.: Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models. J. Am. Stat. Assoc. **115**(530), 866–885 (2020)

Knape, J., De Valpine, P.: Fitting complex population models by combining particle filters with Markov chain Monte Carlo. Ecology **93**(2), 256–263 (2012)

Lewis, J.M., Lakshmivarahan, S., Dhall, S.: Dynamic data assimilation: a least squares approach, vol. 13. Cambridge University Press, Cambridge (2006)

Lim, S., Park, C., Kim, J., Jang, I.: Integrated data assimilation and distance-based model selection with ensemble Kalman filter for characterization of uncertain geological scenarios. Nat. Resour. Res. **29**, 1063–1085 (2020)

Lorenz, E.N.: Predictability: a problem partly solved. In: Proc. seminar on predictability, vol. 1. Reading (1996)

Moradkhani, H., DeChant, C.M., Sorooshian, S.: Evolution of ensemble data assimilation for uncertainty quantification using the particle filter-Markov chain Monte Carlo method. Water Resour. Res. (2012). https://doi.org/10.1029/2012WR012144

Snyder, C., Bengtsson, T., Bickel, P., Anderson, J.: Obstacles to high-dimensional particle filtering. Mon. Weather Rev. **136**(12), 4629–4640 (2008)

Stroud, J.R., Katzfuss, M., Wikle, C.K.: A Bayesian adaptive ensemble Kalman filter for sequential state and parameter estimation. Mon. Weather Rev. **146**(1), 373–386 (2018)

van Leeuwen, P.J.: Nonlinear ensemble data assimilation for the ocean. In: Seminar on recent developments in data assimilation for atmosphere and ocean, ECMWF (2003)

van Leeuwen, P.J., Künsch, H.R., Nerger, L., Potthast, R., Reich, S.: Particle filters for high-dimensional geoscience applications: a review. Q. J. R. Meteorol. Soc. **145**(723), 2335–2365 (2019)

Vetra-Carvalho, S., Van Leeuwen, P.J., Nerger, L., Barth, A., Altaf, M.U., Brasseur, P., Kirchgessner, P., Beckers, J.-M.: State-of-the-art stochastic data assimilation methods for high-dimensional non-gaussian problems. Tellus A Dyn. Meteorol. Oceanogr. **70**(1), 1–43 (2018)

Vrugt, J.A., ter Braak, C.J., Diks, C.G., Schoups, G.: Hydrologic data assimilation using particle Markov chain Monte Carlo simulation: theory, concepts and applications. Adv. Water Resour. **51**, 457–478 (2013)

Wang, X., Verlaan, M., Veenstra, J., Lin, H.X.: Data-assimilation-based parameter estimation of bathymetry and bottom friction coefficient to improve coastal accuracy in a global tide model. Ocean Sci. **18**(3), 881 (2022)

Wiese, T., Rosca, J., Claussen, H.: Reversible jump particle filter (RJPF) for wideband DOA tracking. In: Excursions in harmonic analysis,

Volume 3: The February Fourier talks at the Norbert wiener center, 231–261 (2015)
Xu, K., Wikle, C.K.: Estimation of parameterized Spatio-temporal dynamic models. J. Stat. Plann. Inference **137**(2), 567–588 (2007)

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