Behavior of the iterative ensemble-based variational method in nonlinear problems

Shin'ya Nakano[1,2]

[1]The Institute of Statistical Mathematics, Tachikawa, 190–8562, Japan
[2]Center for Data Assimilation Research and Applications, Joint Support Center for Data Science Research, Tachikawa, Japan.
Correspondence: Shin'ya Nakano (shiny@ism.ac.jp)

Abstract. The behavior of the iterative ensemble-based data assimilation algorithm is discussed. The ensemble-based method for variational data assimilation problems, referred to as the 4-dimensional ensemble variational method (4DEnVar), is a useful tool for data assimilation problems. Although the 4DEnVar is derived based on a linear approximation, highly uncertain problems, where system nonlinearity is significant, are solved by applying this method iteratively. However, the ensemble-based methods basically seek the solution within a lower-dimensional subspace spanned by the ensemble members. It is not necessarily trivial how high-dimensional problems can be solved with the ensemble-based algorithm which employs the lower-dimensional approximation based on the ensemble. In the present study, an ensemble-based iterative algorithm is reformulated to allow us to analyze its behavior in high-dimensional nonlinear problems. The conditions for monotonic convergence to a local maximum of the objective function are discussed in high-dimensional context. It is shown that the ensemble-based algorithm can solve high-dimensional problems by distributing the ensemble in different subspace at each iteration The findings as the results of the present study were also experimentally supported.

1 Introduction

The 4-dimensional ensemble variational method (4DEnVar; Lorenc, 2003; Liu et al., 2008) is a useful tool for practical data assimilation. The 4DEnVar obtains the derivative of the objective function from the approximate Jacobian of a dynamical system model which is estimated by using the ensemble of simulation results. In contrast with the adjoint method, the 4DEnVar does not require an adjoint code which is usually time-consuming to develop. This ensemble method thus allows us to treat the simulation code as a 'black box', and it can easily be implemented.

The 4DEnVar algorithm is derived based on a low-dimensional linear approximation of the high-dimensional nonlinear system model. If the uncertainties in state variables are small, the solution could be found within the range where a linear approximation is valid. However, geophysical systems are often highly uncertain. If the scale of uncertainty is much larger than the range of linearity, a linear approximation would not be justified. In atmospheric applications, uncertainty can usually be reduced by taking sufficient spin-up time. On the other hand, in some geophysical applications, it is difficult to obtain a sufficiently long sequence of observations to allow spin-up. For example, in data assimilation for the interior of the Earth such as lithospheric plates (e.g., Kano et al., 2015) and the outer core (e.g., Sanchez et al., 2019; Minami et al., 2020), time scale of

system dynamics is so long that a sufficient length of an observation sequence is not feasible. It is also difficult to use a long sequence of observations in the Earth's magnetosphere where the amount of observations is limited (e.g., Nakano et al., 2008; Godinez et al., 2016). It is therefore an important issue to consider large uncertainties which could deteriorate the validity of the linear approximation.

Several studies have suggested that estimation in nonlinear problems can be improved by iterative algorithms in which the ensemble is repeatedly updated in each iteration (e.g., Gu and Oliver, 2007; Kalnay and Yang, 2010; Chen and Oliver, 2012; Bocquet and Sakov, 2013, 2014; Raanes et al., 2019). These iterative algorithms can be regarded as a variant of the 4DEnVar method based on an approximation of the Gauss-Newton method or the Levenberg-Marquardt method. The Gauss-Newton and the Levenberg-Marquardt methods are variants of the Newton-Raphson method for solving nonlinear least squares problems by using the Jacobian of a nonlinear function. Thus, when the Gauss-Newton or the Levenberg-Marquardt framework is strictly applied to data assimilation problems, the tangent linear of the system model is required. Indeed, if the tangent linear of the system model is obtained, 4-dimensional variational data assimilation problems can be solved with the incremental formulation (Courtier et al., 1994) which can be regarded as an instance of the Gauss-Newton framework (Lawless et al., 2005). The ensemble-based methods avoid computing the Jacobian of a nonlinear system model by a linear approximation using the ensemble. This ensemble-based approximation is justified if linearity can be assumed over the range where the ensemble members are distributed. However, the ensemble-based methods basically seek the solution within a lower-dimensional subspace spanned by the ensemble members. In many applications in atmospheric sciences, it has been demonstrated that the localization of the covariance matrix is useful for coping with high-dimensional problems (e.g., Buehner, 2005; Liu et al., 2009; Buehner et al., 2010; Yokota et al., 2016). However, it has not necessarily been clarified how general high-dimensional problems, in which the localization of the covariance matrix might not be appropriate, can be solved with the ensemble-based algorithm which employs the lower-dimensional approximation based on the ensemble.

The present study aims to reformulate an ensemble-based iterative algorithm in order to analyze its behavior in high-dimensional nonlinear problems. We then explore the conditions for achieving monotonic convergence to a local maximum of the objective function in high-dimensional nonlinear context. The monotonic convergence means that the discrepancies between estimates and observations are reduced in each iteration. It is ensured that the algorithm would attain a satisfactory result in high-dimensional problems if the ensemble is distributed in a different subspace at each iteration. This study is originally motivated by data assimilation into a geodynamo model to which the author contributed (Minami et al., 2020). However, the present paper focuses on the iterative variational data assimilation algorithm for general uncertain problems in order to avoid the discussion on specific physical processes of geodynamo. In Section 2, the formulation of the variational data assimilation problem is described. In Section 3, the basic idea of the ensemble variational method is explained. The iterative version is introduced as an algorithm for maximizing the log-likelihood function in Section 4, and the behavior of the iterative algorithm is evaluated in Section 5. In Section 6, a Bayesian extension is introduced. Section 7 experimentally verifies our findings. Finally, a discussion and conclusions are presented in Section 8.

2 4-dimensional variational data assimilation (4DEnVar)

In the following, the system state at time t_k is denoted as $\boldsymbol{x}_k$ and the observation at t_k is denoted as $\boldsymbol{y}_k$. We consider a strong-constraint data assimilation problem where the evolution of state $\boldsymbol{x}_k$ is given by

$$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \tag{1}$$

and the relation between $\boldsymbol{y}_k$ and $\boldsymbol{x}_k$ is written in the following form:

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{2}$$

where $\boldsymbol{w}_k$ indicates the observation noise. Assuming that $\boldsymbol{w}_k$ obeys a Gaussian distribution with mean $\mathbf{0}$ and covariance matrix R_k, then

$$p(\boldsymbol{w}_k) \propto \exp\left[-\frac{1}{2}\boldsymbol{w}_k^T \mathsf{R}_k^{-1} \boldsymbol{w}_k\right]. \tag{3}$$

The likelihood of $\boldsymbol{x}_k$ given $\boldsymbol{y}_k$ is

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) \propto \exp\left[-\frac{1}{2}\left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right)^T \mathsf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right)\right]. \tag{4}$$

Since we assume a deterministic system as stated in Eq. (1), $\boldsymbol{h}_k(\boldsymbol{x}_k)$ can be written as a function of an initial value $\boldsymbol{x}_0$ as

$$\boldsymbol{h}_k(\boldsymbol{x}_k) = \boldsymbol{g}_k(\boldsymbol{x}_0), \tag{5}$$

where $\boldsymbol{g}_k$ is the following composite function

$$\boldsymbol{g}_k(\boldsymbol{x}_0) = \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \boldsymbol{f}_{k-1} \circ \cdots \boldsymbol{f}_1(\boldsymbol{x}_0). \tag{6}$$

The likelihood in Eq. (4) is then written as

$$p(\boldsymbol{y}_k|\boldsymbol{x}_0) \propto \exp\left[-\frac{1}{2}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right)^T \mathsf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right)\right]. \tag{7}$$

When the prior distribution of $\boldsymbol{x}_0$ is assumed to be Gaussian with mean $\bar{\boldsymbol{x}}_{0,b}$ and covariance matrix $\mathsf{P}_{0,b}$ defined by

$$p(\boldsymbol{x}_0) \propto \exp\left[-\frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\right)^T \mathsf{P}_{0,b}^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\right)\right], \tag{8}$$

the Bayesian posterior distribution of $\boldsymbol{x}_0$ given the whole sequence of observations from t_1 to t_K, $\boldsymbol{y}_{1:K}$, can be obtained as follows:

$$p(\boldsymbol{x}_0|\boldsymbol{y}_{1:K}) \propto \exp\left[-\frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\right)^T \mathsf{P}_{0,b}^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\right) - \frac{1}{2}\sum_{k=1}^{K}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right)^T \mathsf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right)\right]. \tag{9}$$

The maximum of the posterior can be found by maximizing the following objective function:

$$J(\boldsymbol{x}_0) = -\frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\right)^T \mathsf{P}_{0,b}^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\right) - \frac{1}{2}\sum_{k=1}^{K}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right)^T \mathsf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right). \tag{10}$$

3 Ensemble-based method

The maximization of the objective function J is conventionally performed by the adjoint method, which differentiates J based on the adjoint matrix of the Jacobian of the function $\boldsymbol{f}_k$ in Eq. (1). For a practical high-dimensional simulation model, however, it is an extremely laborious task to develop the adjoint code which represents the adjoint matrix of the Jacobian of the forward simulation model. The 4DEnVar is an alternative method for obtaining an approximate maximum of J without using the adjoint code. The 4DEnVar employs an ensemble of N simulation results $\{\boldsymbol{x}_{0:K}^{(1)}, \ldots, \boldsymbol{x}_{0:K}^{(N)}\}$, where $\boldsymbol{x}_{0:K}$ indicates the whole sequence of the states from t_0 to t_K; that is, $\boldsymbol{x}_{0:K} = (\boldsymbol{x}_0^T \ \cdots \ \boldsymbol{x}_K^T)^T$. The initial state of each ensemble member $\boldsymbol{x}_0^{(i)}$ is assumed to be sampled from the Gaussian distribution $\mathcal{N}(\boldsymbol{x}_0; \boldsymbol{x}_{0,b}, \mathsf{P}_{0,b})$. The objective function in Eq. (10) is approximated by using this ensemble.

For convenience, we define the following matrix $\mathsf{X}_{0,b}$ from the initial states of ensemble members:

$$\mathsf{X}_{0,b} = \frac{1}{\sqrt{N}} \left(\begin{array}{ccc} \boldsymbol{x}_0^{(1)} - \bar{\boldsymbol{x}}_{0,b} & \cdots & \boldsymbol{x}_0^{(N)} - \bar{\boldsymbol{x}}_{0,b} \end{array} \right). \tag{11}$$

Assuming that the optimal $\boldsymbol{x}_0$ can be written as a linear combination of the ensemble members, we can write $\boldsymbol{x}_0$ in the following form:

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_{0,b} + \mathsf{X}_{0,b}\boldsymbol{w}. \tag{12}$$

This assumption means that $\boldsymbol{x}_0$ is within the subspace spanned by the ensemble members. The quality of an estimate with the 4DEnVar can thus be poor if there are insufficient ensemble members. In practical applications of the 4DEnVar, a localization technique is usually used to avoid this problem (e.g., Buehner, 2005; Liu et al., 2009; Buehner et al., 2010; Yokota et al., 2016). However, the present paper does not consider localization because the focus here is on the basic behavior of the 4DEnVar. If we assume that the rank of $\mathsf{X}_{0,b}$ is N ($< \dim \boldsymbol{x}_0$) and approximate the inverse of $\mathsf{P}_{0,b}$ by the Moore-Penrose inverse matrix of $\mathsf{X}_{0,b}\mathsf{X}_{0,b}^T$, the first term of the right-hand side of Eq. (10) can be approximated as

$$-\frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\right)^T \mathsf{P}_{0,b}^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\right) = -\frac{1}{2}\boldsymbol{w}^T\mathsf{X}_{0,b}^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_{0,b}\boldsymbol{w} \approx -\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}. \tag{13}$$

This corresponds to a low-rank approximation within the subspace spanned by the ensemble members. The prior mean $\bar{\boldsymbol{x}}_{0,b}$ is usually given by the ensemble mean of $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$. In such a case, it is necessary to ignore the subspace along the vector $\mathbf{1} = (1 \ \cdots \ 1)^T$ to reach the approximation of Eq. (13). The function $\boldsymbol{g}_k(\boldsymbol{x}_0)$ is approximated based on the first-order Taylor expansion:

$$\boldsymbol{g}_k(\boldsymbol{x}_0) \approx \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) + \mathsf{G}_k(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}) \approx \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) + \mathsf{G}_k\mathsf{X}_{0,b}\boldsymbol{w}, \tag{14}$$

where G_k is the Jacobian of $\boldsymbol{g}_k$ at $\bar{\boldsymbol{x}}_{0,b}$. The matrix $\mathsf{G}_k\mathsf{X}_{0,b}$ in Eq. (14) is approximated as

$$\mathsf{G}_k\mathsf{X}_{0,b} \approx \frac{1}{\sqrt{N}}\left(\boldsymbol{g}_k(\boldsymbol{x}_0^{(1)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) \quad \cdots \quad \boldsymbol{g}_k(\boldsymbol{x}_0^{(N)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b})\right). \tag{15}$$

Defining the right-hand side of Eq. (15) as Γ_k, that is,

$$\Gamma_k = \frac{1}{\sqrt{N}}\left(\boldsymbol{g}_k(\boldsymbol{x}_0^{(1)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) \quad \cdots \quad \boldsymbol{g}_k(\boldsymbol{x}_0^{(N)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b})\right) \approx \mathsf{G}_k\mathsf{X}_{0,b}, \tag{16}$$

we obtain a further approximation of the function $\boldsymbol{g}_k(\boldsymbol{x}_0)$ in Eq. (14):

$$\boldsymbol{g}_k(\boldsymbol{x}_0) \approx \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) + \Gamma_k \boldsymbol{w} \tag{17}$$

(e.g., Zupanski et al., 2008; Bannister, 2017). Using Eqs. (13) and (17), the objective function in Eq. (10) can be approximated as a function of $\boldsymbol{w}$ as follows:

$$\hat{J}_w(\boldsymbol{w}) = -\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) - \Gamma_k\boldsymbol{w})\,\mathsf{R}_k^{-1}\,(\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) - \Gamma_k\boldsymbol{w}) \tag{18}$$

where we defined $\hat{J}_w(\boldsymbol{w}) = J(\bar{\boldsymbol{x}}_{0,b} + \mathsf{X}_{0,b}\boldsymbol{w})$.

The approximate objective function $\hat{J}_w$ is a quadratic function of $\boldsymbol{w}$ and it no longer contains the Jacobian of the function $\boldsymbol{g}_k$. The maximization of $\hat{J}_w$ is thus much easier than that of the original objective function in Eq. (10). The derivative of $\hat{J}_w$ with respect to $\boldsymbol{w}$ becomes

$$\nabla_w \hat{J}_w = \boldsymbol{w} - \sum_{k=1}^{K}\left(\Gamma_k^T\mathsf{R}_k^{-1}\left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) - \Gamma_k\boldsymbol{w}\right]\right) \tag{19}$$

(Liu et al., 2008). The Hessian matrix of $\hat{J}_w$ is then obtained as

$$\mathsf{H}_{\hat{J}_w} = \mathbf{I} + \sum_k [\Gamma_k^T\mathsf{R}_k^{-1}\Gamma_k]. \tag{20}$$

We can thus immediately find the value of $\boldsymbol{w}$ maximizing $\hat{J}_w$:

$$\hat{\boldsymbol{w}} = \left(\mathbf{I} + \sum_k \Gamma_k^T\mathsf{R}_k^{-1}\Gamma_k\right)^{-1}\sum_k\left(\Gamma_k^T\mathsf{R}_k^{-1}\left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b})\right]\right). \tag{21}$$

Inserting $\hat{\boldsymbol{w}}$ into Eq. (12), we obtain an estimate of $\boldsymbol{x}_0$ as follows:

$$\hat{\boldsymbol{x}}_0 = \bar{\boldsymbol{x}}_{0,b} + \mathsf{X}_{0,b}\hat{\boldsymbol{w}}. \tag{22}$$

This solution in Eq. (22) is similar to the ensemble Kalman smoother (van Leeuwen and Evensen, 1996; Evensen and van Leeuwen, 2000) although the whole sequence of observations is referred to in Eq. (21). Even if a large amount of data are used, it would not seriously affect the computational cost because the computation of the inverse matrix can be conducted in N-dimensional space. This is also an advantage of the ensemble-based method.

4 Iterative algorithm

Since Eqs. (21) and (22) do not require the Jacobian of the function $\boldsymbol{g}_k$, it can be applied as a post-process provided that an ensemble of the simulation runs is prepared in advance. However, this solution, which maximizes the objective function in Eq. (18), relies on Eq. (15) which approximates matrix $\mathsf{G}_k\mathsf{X}_{0,b}$ by using the ensemble. This approximation is based on the first-order

approximation shown in Eq. (14). Where $\boldsymbol{x}_0$ exhibits high uncertainty and $\|\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b}\|$ can be large, this approximation appears to be invalid. Therefore, it is not guaranteed that the estimate with Eq. (22) provides the optimal $\boldsymbol{x}_0$ which maximizes the original log-posterior density function in Eq. (10) even if we accept that the solution is limited within the ensemble subspace.

Where the initially prepared ensemble is used, it is unlikely that a better solution than Eq. (22) could be achieved. We then consider an iterative algorithm which generates a new ensemble based on the previous estimate in each iteration. The algorithm introduced in the following is basically the same as the method referred to as the iterative ensemble Kalman filter (Bocquet and Sakov, 2013, 2014), but we employ a formulation to allows evaluation of the behavior and a slight extension. To derive an algorithm analogous to that in the previous section, we at first consider the following log-likelihood function:

$$J_\ell(\boldsymbol{x}_0) = -\frac{1}{2}\sum_{k=1}^{K}[\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)]^T \mathsf{R}^{-1}[\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)]. \tag{23}$$

instead of the log-posterior density function in Eq. (10). Maximization of the Bayesian-type objective function in Eq. (10) will be discussed in Section 6.

In the following, we combine the vectors of the whole time sequence from t_1 to t_K into one single vector; that is, $\boldsymbol{y} = \boldsymbol{y}_{1:K}$ and $\boldsymbol{g}(\boldsymbol{x}_0) = \boldsymbol{g}_{0:K}(\boldsymbol{x}_0)$. The covariance matrices $\mathsf{R}_1, \ldots, \mathsf{R}_K$ are also combined into one block diagonal matrix R which satisfies

$$\boldsymbol{y}^T\mathsf{R}^{-1}\boldsymbol{y} = \sum_{k=1}^{K}\boldsymbol{y}_k^T\mathsf{R}_k^{-1}\boldsymbol{y}_k. \tag{24}$$

Accordingly, the log-likelihood function of Eq. (23) is rewritten as

$$J_\ell(\boldsymbol{x}_0) = -\frac{1}{2}[\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}_0)]^T \mathsf{R}^{-1}[\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}_0)]. \tag{25}$$

In our iterative algorithm, the m-th step starts with an ensemble of initial values $\{\boldsymbol{x}_{0,m-1}^{(1)}, \ldots, \boldsymbol{x}_{0,m-1}^{(N)}\}$ obtained in the neighbor of the $(m-1)$-th estimate $\bar{\boldsymbol{x}}_{0,m-1}$. Typically, the ensemble is generated so that the ensemble mean is equal to $\bar{\boldsymbol{x}}_{0,m-1}$; that is,

$$\bar{\boldsymbol{x}}_{0,m-1} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_{0,m-1}^{(i)}, \tag{26}$$

although it is not necessary to satisfy this equation. A simulation run initialized at $\boldsymbol{x}_{0,m-1}^{(i)}$ yields $\boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(i)})$, and we obtain the ensemble of the simulation results $\{\boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(1)}), \ldots, \boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(N)})\}$. Defining the matrices

$$\mathsf{X}_{m-1} = \frac{1}{\sqrt{N}}\left(\boldsymbol{x}_{0,m-1}^{(1)} - \bar{\boldsymbol{x}}_{0,m-1} \quad \cdots \quad \boldsymbol{x}_{0,m-1}^{(N)} - \bar{\boldsymbol{x}}_{0,m-1}\right), \tag{27}$$

$$\mathsf{\Gamma}_{m-1} = \frac{1}{\sqrt{N}}\left(\boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(1)}) - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \quad \cdots \quad \boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(N)}) - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right), \tag{28}$$

we consider the following m-th objective function:

$$\begin{aligned}&\check{J}_{\ell,m}(\boldsymbol{w}_m|\bar{\boldsymbol{x}}_{0,m-1})\\ &\quad = -\frac{\sigma_m^2}{2}\boldsymbol{w}_m^T\boldsymbol{w}_m - \frac{1}{2}[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) - \mathsf{\Gamma}_{m-1}\boldsymbol{w}_m]^T \mathsf{R}^{-1}[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) - \mathsf{\Gamma}_{m-1}\boldsymbol{w}_m].\end{aligned} \tag{29}$$

where σ_m is an appropriately chosen parameter. This objective function $\check{J}_{\ell,m}$ is maximized when

$$\hat{\boldsymbol{w}}_m = \left(\sigma_m^2 \mathbf{I} + \Gamma_{m-1}^T \mathsf{R}^{-1} \Gamma_{m-1}\right)^{-1} \left(\Gamma_{m-1}^T \mathsf{R}^{-1} [\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})]\right), \tag{30}$$

and $\bar{\boldsymbol{w}}_m$ provides the m-th estimate of $\bar{\boldsymbol{x}}_{0,m}$ as follows:

$$\bar{\boldsymbol{x}}_{0,m} = \bar{\boldsymbol{x}}_{0,m-1} + \mathsf{X}_{m-1} \bar{\boldsymbol{w}}_m. \tag{31}$$

Unless converged, members of the next ensemble are generated in the neighbor of $\bar{\boldsymbol{x}}_{0,m}$ so that $\|\boldsymbol{x}_{0,m}^{(i)} - \bar{\boldsymbol{x}}_{0,m}\|^2$ is small for each i, and we proceed to the next iteration. By iterating the above procedures until convergence, the optimal $\hat{\boldsymbol{x}}_0$ which maximizes J_ℓ is attained.

The form of $\check{J}_{\ell,m}$ in Eq. (29) looks similar to that of $\hat{J}_w$ in Eq. (18). However, the meaning of the first term of Eq. (29) is different from that of the first term of Eq. (18). The first term of Eq. (18) corresponded to the Bayesian prior. On the other hand, the first term of Eq. (29) is a penalty term to ensure monotonic convergence as explained later. After iterations until convergence, the contribution of this penalty term would decay, and the log-likelihood function in Eq. (23) is maximized in the end.

We can consider various ways to obtain an ensemble satisfying Eq. (26). Bocquet and Sakov (2013) proposed to obtain a matrix X_m as a scalar multiple of $\mathsf{X}_{0,b}$:

$$\mathsf{X}_m = \alpha_m \mathsf{X}_{0,b} \quad (\alpha_m \geq 0). \tag{32}$$

where $\mathsf{X}_{0,b}$ is a matrix defined in Eq. (11). A new ensemble for next iteration is generated to satisfy

$$\mathsf{X}_m = \frac{1}{\sqrt{N}} \left(\boldsymbol{x}_{0,m}^{(1)} - \bar{\boldsymbol{x}}_{0,m} \quad \cdots \quad \boldsymbol{x}_{0,m}^{(N)} - \bar{\boldsymbol{x}}_{0,m} \right). \tag{33}$$

As discussed later, α_m should be taken to be so small that a linear approximation is valid over the range of ensemble dispersion. The value of α_m can be fixed at a small value. Otherwise, α_m may be reduced gradually in each iteration so that the spread of the ensemble eventually becomes small. We can also shrink the ensemble by using a similar scheme to the ensemble transform Kalman filter (Bishop et al., 2001; Livings et al., 2008) which obtains X_m as outlined by Bocquet and Sakov (2012); that is,

$$\mathsf{X}_m = \mathsf{X}_{m-1} \mathsf{T}_m, \tag{34}$$

where T_m is the ensemble transform matrix given as

$$\mathsf{T}_m = \mathsf{U}_m (\mathbf{I} + \Lambda_m)^{-\frac{1}{2}} \mathsf{U}_m^T. \tag{35}$$

In Eq. (35), $\mathbf{I}$ is the identity matrix and $\mathsf{U}_m \Lambda_m \mathsf{U}_m^T$ is the eigenvalue decomposition of the matrix $\sigma_m^{-2} \Gamma_{m-1}^T \mathsf{R}^{-1} \Gamma_{m-1}$, where U_m is an orthogonal matrix consisting of the eigenvectors, and the matrix Λ_m is a diagonal matrix of the eigenvalues.

If the ensemble is updated according to Eq. (32) or (34), the estimate of $\boldsymbol{x}_0$ is constrained within the subspace spanned by the initial ensemble members $\{\boldsymbol{x}_{0,0}^{(1)}, \ldots, \boldsymbol{x}_{0,0}^{(N)}\}$. We can avoid confining the ensemble within a subspace by randomly generating ensemble members from a Gaussian distribution with mean $\bar{\boldsymbol{x}}_{0,m}$ and variance Q_m as:

$$\boldsymbol{x}_{0,m}^{(i)} \sim \mathcal{N}(\bar{\boldsymbol{x}}_{0,m}, \mathsf{Q}_m), \quad (i = 1, \ldots, N). \tag{36}$$

Although this method has a limitation when applying it to Bayesian estimation as explained later, it would be effective if applicable.

The iterative algorithm is summarized in Algorithm 1. The procedures in this iterative algorithm are similar to those in the ensemble-based multiple data assimilation method (Emerick and Reynolds, 2012, 2013), which aims to obtain the maximum of the Bayesian posterior function, especially if the ensemble is updated with Eq. (34). The multiple data assimilation method does not perform iterations until convergence, but it performs iterations only a few times to estimate the maximum of the posterior although it can provide a biased solution in nonlinear problems (Evensen, 2018). In order to achieve the convergence to the maximum of the Bayesian posterior in our framework, the objective function in each iteration should be modified as discussed in Section 6.

Algorithm 1 Iterative algorithm for maximizing the log-likelihood function J_ℓ.

Give an initial estimate of $\boldsymbol{x}_0$, $\bar{\boldsymbol{x}}_{0,0}$.
Give an initial square root of the covariance, X_0.
Let $m = 1$.
while unconverged **do**
  Generate an ensemble $\{\boldsymbol{x}_{0,m-1}^{(i)}\}_{i=1}^N$ in the neighbor of $\bar{\boldsymbol{x}}_{0,m-1}$ by using either of Eq. (32), (34), or (36).
  Obtain X_{m-1} and $\Gamma_{k,m-1}$ in Eqs. (27) and (28).
  Compute $\hat{\boldsymbol{w}}_m$ in Eq. (30).
  Compute the m-th mean vector $\bar{\boldsymbol{x}}_{0,m}$ according to Eq. (31).
  Compute the matrix X_m according to Eq. (34).
  Let $m := m + 1$
end while

5 Rationale of the algorithm

Eq. (30) can be regarded as an approximation of the Levenberg-Marquardt method (e.g., Nocedal and Wright, 2006) for maximizing the log-likelihood function in Eq. (23) within the subspace spanned by $\{\boldsymbol{x}_{0,0}^{(1)}, \ldots, \boldsymbol{x}_{0,0}^{(N)}\}$. In particular, if σ_m^2 is zero, Eq. (30) can be regarded as an approximation of the Gauss-Newton method. Indeed, Bocquet and Sakov (2013, 2014) derived a similar algorithm as an approximation of the Levenberg-Marquardt method or the Gauss-Newton method. However, the Levenberg-Marquardt method basically requires the Jacobian of the function $\boldsymbol{g}_k$, G_{m-1}. Since the above iterative algorithm does not directly use G_{m-1}, it would not be trivial how the convergence of this algorithm is achieved. This issue is explored in this section.

We hereinafter assume that $\boldsymbol{g}(\boldsymbol{x}_0)$ is at least twice differentiable. The Taylor expansion up to the second-order term of J_ℓ becomes

$$\begin{aligned} J_\ell(\boldsymbol{x}_0) = &-\frac{1}{2}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T \mathsf{R}^{-1}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right] + \left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T \mathsf{R}^{-1}\mathsf{G}_{m-1}(\boldsymbol{x}_0-\bar{\boldsymbol{x}}_{0,m-1}) \\ &-\frac{1}{2}(\boldsymbol{x}_0-\bar{\boldsymbol{x}}_{0,m-1})^T\mathsf{G}_{m-1}^T\mathsf{R}^{-1}\mathsf{G}_{m-1}(\boldsymbol{x}_0-\bar{\boldsymbol{x}}_{0,m-1}) \\ &+\frac{1}{4}(\boldsymbol{x}_0-\bar{\boldsymbol{x}}_{0,m-1})^T\left[\left(\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right)^T\mathsf{R}^{-1}\left(\nabla^2\boldsymbol{g}\right)\right](\boldsymbol{x}_0-\bar{\boldsymbol{x}}_{0,m-1}) \\ &+O(\|\boldsymbol{x}_0-\bar{\boldsymbol{x}}_{0,m-1}\|^3), \end{aligned} \tag{37}$$

where G_{m-1} is the Jacobian at $\bar{\boldsymbol{x}}_{0,m-1}$ and $\left(\nabla^2\boldsymbol{g}\right)$ is a third-order tensor which consists of the Hessian matrix of each element of the vector-valued function $\boldsymbol{g}(\boldsymbol{x}_0)$. As done in Eq. (12), we assume

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_{0,m-1} + \mathsf{X}_{m-1}\boldsymbol{w}_m, \tag{38}$$

where X_{m-1} is obtained by Eq. (27) given the ensemble $\{\boldsymbol{x}_{0,m-1}^{(1)},\ldots,\boldsymbol{x}_{0,m-1}^{(N)}\}$. We then have

$$\begin{aligned} J_\ell(\boldsymbol{x}_0) &= J_\ell(\bar{\boldsymbol{x}}_{0,m-1}+\mathsf{X}_{m-1}\boldsymbol{w}_m) \\ &= -\frac{1}{2}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T\mathsf{R}^{-1}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right] + \left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T\mathsf{R}^{-1}\mathsf{G}_{m-1}\mathsf{X}_{m-1}\boldsymbol{w}_m \\ &\quad -\frac{1}{2}\boldsymbol{w}_m^T\mathsf{X}_{m-1}^T\mathsf{G}_{m-1}^T\mathsf{R}^{-1}\mathsf{G}_{m-1}\mathsf{X}_{m-1}\boldsymbol{w}_m \\ &\quad +\frac{1}{4}\boldsymbol{w}_m^T\mathsf{X}_{m-1}^T\left[\left(\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right)^T\mathsf{R}^{-1}\left(\nabla^2\boldsymbol{g}\right)\right]\mathsf{X}_{m-1}\boldsymbol{w}_m + O(\|\boldsymbol{w}_m\|^3). \end{aligned} \tag{39}$$

In practical cases, the Jacobian matrix G_{m-1} is typically unavailable. Ensemble variational methods thus employ the first-order approximation in Eq. (15) for $\mathsf{G}_{m-1}\mathsf{X}_{m-1}$; that is,

$$\mathsf{G}_{m-1}\mathsf{X}_{m-1} \approx \mathsf{\Gamma}_{m-1}, \tag{40}$$

where

$$\mathsf{\Gamma}_{m-1} = \frac{1}{\sqrt{N}}\left(\boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(1)})-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \quad \cdots \quad \boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(N)})-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right). \tag{41}$$

To evaluate this approximation when $\boldsymbol{x}_0$ has a large uncertainty, we consider the following expansion of $\boldsymbol{g}(\boldsymbol{x}_0)$ for each ensemble member $\boldsymbol{x}_0^{(i)}$:

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{x}_0^{(i)}) = &\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) + \mathsf{G}_{m-1}(\boldsymbol{x}_0^{(i)}-\bar{\boldsymbol{x}}_{0,m-1}) \\ &+\frac{1}{2}(\boldsymbol{x}_0^{(i)}-\bar{\boldsymbol{x}}_{0,m-1})^T\left(\nabla^2\boldsymbol{g}\right)(\boldsymbol{x}_0^{(i)}-\bar{\boldsymbol{x}}_{0,m-1}) + O(\|\boldsymbol{x}_0^{(i)}-\bar{\boldsymbol{x}}_{0,m-1}\|^3). \end{aligned} \tag{42}$$

If we consider a vector $\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m$, it becomes

$$
\begin{aligned}
&\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m \\
&= \frac{1}{\sqrt{N}}\sum_{i=1}^{N} w^{(i)}\mathsf{G}_{m-1}(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1}) \\
&\quad + \frac{1}{2\sqrt{N}}\sum_{i=1}^{N} w^{(i)}(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1})^T\left(\nabla^2\boldsymbol{g}\right)(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1}) \\
&= \mathsf{G}_{m-1}\mathsf{X}_{m-1}\boldsymbol{w}_m \\
&\quad + \frac{1}{2\sqrt{N}}\sum_{i=1}^{N} w^{(i)}\left[(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1})^T\left(\nabla^2\boldsymbol{g}\right)(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1}) + O(\|\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1}\|^3)\right].
\end{aligned} \tag{43}
$$

If $\mathsf{G}_{m-1}\mathsf{X}_{m-1}\boldsymbol{w}_m$, which is contained in the first-order term in Eq. (39), is approximated by $\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m$, this means that the second- and higher-order terms of the right-hand side of Eq. (43) are neglected. Indeed, this can be justified if the spread of the ensemble is taken to be small. In our iterative scheme, the ensemble spread can be tuned freely. Even if the scale of $\|\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1}\|$ is very small, any $\boldsymbol{x}_0$, which may have a large uncertainty, can be represented by taking the scale of $\|\boldsymbol{w}_m\|$ to be large according to Eq. (38). The nonlinear terms of the right-hand side of Eq. (43) are of the order of $\|\boldsymbol{w}_m\|$, while they are of the order of $\|\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1}\|^2$ or higer order. Thus, if the spread of the ensemble is taken to be small, the nonlinear terms of Eq. (43) would be suppressed, and we obtain

$$
\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m \approx \mathsf{G}_{m-1}\mathsf{X}_{m-1}\boldsymbol{w}_m. \tag{44}
$$

Consequently, we can apply the approximation in Eq. (40) to Eq. (39). Defining a function $J_{\ell,w_m}(\boldsymbol{w}_m)$ as

$$
\begin{aligned}
J_{\ell,w_m}(\boldsymbol{w}_m) = &-\frac{1}{2}\left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T\mathsf{R}^{-1}\left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right] + \left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T\mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m \\
&- \frac{1}{2}\boldsymbol{w}_m^T\mathsf{\Gamma}_{m-1}^T\mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m \\
&+ \frac{1}{4}\boldsymbol{w}_m^T\mathsf{X}_{m-1}^T\left[(\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}))^T\mathsf{R}^{-1}\left(\nabla^2\boldsymbol{g}\right)\right]\mathsf{X}_{m-1}\boldsymbol{w}_m + O(\|\boldsymbol{w}_m\|^3),
\end{aligned} \tag{45}
$$

$J_{\ell,w_m}(\boldsymbol{w}_m)$ gives an approximation of $J_\ell(\bar{\boldsymbol{x}}_{0,m-1} + \mathsf{X}_{m-1}\boldsymbol{w}_m)$ in Eq. (39):

$$
J_\ell(\bar{\boldsymbol{x}}_{0,m-1} + \mathsf{X}_{m-1}\boldsymbol{w}_m) \approx J_{\ell,w_m}(\boldsymbol{w}_m). \tag{46}
$$

The fourth term on the right-hand side of Eq. (45) would not necessarily be suppressed even if the ensemble variance were taken to be small, because it is of the order of $\|\boldsymbol{w}_m\|^2$ and of the order of $\|\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_{0,m-1}\|^2$. To control the effect of this term, we introduce the idea of the minorize-maximize algorithm (MM algorithm) (Lange et al., 2000; Lange, 2016). The MM algorithm is a class of iterative algorithms which considers a surrogate function which minorizes the objective function $\phi(\boldsymbol{z})$ and maximizes the surrogate function. Although the Levenberg-Marquardt method can also be regarded as an instance of the MM algorithm, the generic idea of the MM algorithm gives a striking insight into the behavior of the algorithm.

At the m-th step of the MM algorithm, the surrogate function given the $(m-1)$-th estimate $\boldsymbol{z}_{m-1}$, $\psi(\boldsymbol{z}_0|\boldsymbol{z}_{m-1})$, is chosen to satisfy the following conditions:

$$\psi(\boldsymbol{z}|\boldsymbol{z}_{m-1}) \leq \phi(\boldsymbol{z}), \tag{47a}$$

$$\psi(\boldsymbol{z}_{m-1}|\boldsymbol{z}_{m-1}) = \phi(\boldsymbol{z}_{m-1}). \tag{47b}$$

The m-th estimate, $\boldsymbol{z}_m$, is obtained by maximizing the m-th surrogate function, $\psi(\boldsymbol{z}|\boldsymbol{z}_{m-1})$. Since $\boldsymbol{z}_m$ obviously satisfies

$$\phi(\boldsymbol{z}_{m-1}) = \psi(\boldsymbol{z}_{m-1}|\boldsymbol{z}_{m-1}) \leq \psi(\boldsymbol{z}_m|\boldsymbol{z}_{m-1}) \leq \phi(\boldsymbol{z}_m) \tag{48}$$

it is guaranteed that the m-th estimate is as good as or better than the $(m-1)$-th estimate. After iterations, $\boldsymbol{z}_m$ converges to a stationary point $\boldsymbol{z}_s$ of the objective function $\phi(\boldsymbol{z})$ (Lange, 2016). If the Hessian matrix of $\phi(\boldsymbol{z})$ is negative definite in a neighborhood of $\boldsymbol{z}_s$, the stationary point $\boldsymbol{z}_s$ becomes a local maximum (e.g., Nocedal and Wright, 2006). Therefore, the estimate would monotonically converge to a local maximum of $\phi(\boldsymbol{z})$ by repeating iterations if

- the surrogate function $\psi(\boldsymbol{z}|\boldsymbol{z}_m)$ is twice differentiable and satisfies Eqs. (47a) and (47b),
- and the Hessian of $\phi(\boldsymbol{z})$ is negative definite in a neighborhood of the stationary point $\boldsymbol{z}_s$.

Here we consider the following surrogate function $J^\dagger_{\ell,w_m}$:

$$\begin{aligned}
&J^\dagger_{\ell,w_m}(\boldsymbol{w}_m|\bar{\boldsymbol{x}}_{0,m-1}) \\
&\quad = -\frac{\sigma_m^2}{2}\boldsymbol{w}_m^T\boldsymbol{w}_m - \frac{1}{2}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T \mathsf{R}^{-1}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right] \\
&\qquad + \left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T \mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m - \frac{1}{2}\boldsymbol{w}_m^T\mathsf{\Gamma}_{m-1}^T\mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m \\
&\quad = -\frac{\sigma_m^2}{2}\boldsymbol{w}_m^T\boldsymbol{w}_m - \frac{1}{2}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})-\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m\right]^T \mathsf{R}^{-1}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})-\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m\right],
\end{aligned} \tag{49}$$

which is a similar treatment to Böhning and Lindsay (1988). For a given Δ, we can take σ_m^2 so that the following inequality holds over $\|\boldsymbol{w}_m\| < \Delta$:

$$-\frac{\sigma_m^2}{2}\boldsymbol{w}_m^T\boldsymbol{w}_m \leq \frac{1}{4}\boldsymbol{w}_m^T\mathsf{X}_{m-1}^T\left[\left(\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right)^T\mathsf{R}^{-1}\left(\nabla^2\boldsymbol{g}\right)\right]\mathsf{X}_{m-1}\boldsymbol{w}_m + O(\|\boldsymbol{w}_m\|^3). \tag{50}$$

where equality holds if $\boldsymbol{w}_m = \boldsymbol{0}$. If σ_m^2 is chosen so that the inequality (50) is satisfied, $J^\dagger_{\ell,w}$ satisfies the followings:

$$J^\dagger_{\ell,w_m}(\boldsymbol{w}_m|\bar{\boldsymbol{x}}_{0,m-1}) \leq J_{\ell,w_m}(\boldsymbol{w}_m) \approx J_\ell(\bar{\boldsymbol{x}}_{0,m-1} + \mathsf{X}_{m-1}\boldsymbol{w}_m), \tag{51}$$

$$J^\dagger_{\ell,w_m}(\boldsymbol{0}|\bar{\boldsymbol{x}}_{0,m-1}) = J_{\ell,w_m}(\boldsymbol{0}) = J_\ell(\bar{\boldsymbol{x}}_{0,m-1}). \tag{52}$$

This means that $J^\dagger_{\ell,w_m}$ can be used as a surrogate function for maximizing $J_{\ell,w_m}(\boldsymbol{w}_m)$ according to the MM algorithm. Since Eq. (49) is the same as Eq. (29), the maximum of $J^\dagger_{\ell,w_m}$ is achieved when $\boldsymbol{w}_m = \hat{\boldsymbol{w}}_m$ where $\hat{\boldsymbol{w}}_m$ is given by Eq. (30). Obviously, $\hat{\boldsymbol{w}}_m$ satisfies the following inequality:

$$J^\dagger_{\ell,w_m}(\boldsymbol{0}|\bar{\boldsymbol{x}}_{0,m-1}) \leq J^\dagger_{\ell,w_m}(\hat{\boldsymbol{w}}_m|\bar{\boldsymbol{x}}_{0,m-1}), \tag{53}$$

and therefore, if the approximation in Eq. (40) is valid, we obtain the following result:

$$J_\ell(\bar{\boldsymbol{x}}_{0,m-1}) = J_{\ell,w_m}(\mathbf{0}) \leq J_{\ell,w_m}(\hat{\boldsymbol{w}}_m) \approx J_\ell(\bar{\boldsymbol{x}}_{0,m-1} + \mathsf{X}_{m-1}\hat{\boldsymbol{w}}_m) = J_\ell(\bar{\boldsymbol{x}}_{0,m}), \tag{54}$$

where $\bar{\boldsymbol{x}}_{0,m}$ is given by Eq. (31).

The above discussion is valid regardless of the choice of the ensemble $\{\boldsymbol{x}_{0,m}^{(1)}, \ldots, \boldsymbol{x}_{0,m}^{(N)}\}$ in each iteration as far as the approximation in Eq. (40) is applicable. This suggests we can use various ways to update the ensemble, including Eq. (36) which does not confine the ensemble within a particular subspace. It should be noted that the equality of Eq. (53) holds at a stationary point in the subspace spanned by the ensemble members. If the update of the ensemble in each iteration is carried out with Eq. (32) or (34), the ensemble is confined within a particular subspace spanned by the initial ensemble, and $\bar{\boldsymbol{x}}_{0,m-1}$ would converge to a stationary point in this subspace. According to Eq. (37), if the nonlinearity of $\boldsymbol{g}$ is not severe when $\left(\nabla^2 \boldsymbol{g}\right)$ is not dominant, the Hessian of J_ℓ is negative definite in a region where $\|\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\|$ is small enough. This suggests that the iterative algorithm in Section 4 would attain at least a local maximum of J_ℓ in the subspace for weakly nonlinear problems if Eq. (36) is applicable. If the ensemble is updated according to Eq. (36), a stationary point is sought in a different subspace in each iteration. If Q_m is full rank, J_ℓ would increase until a point which can be regarded as a stationary point in any N-dimensional subspace, and $\bar{\boldsymbol{x}}_{0,m-1}$ would thus converge to a local minimum in the full vector space after infinite iterations.

Based on the foregoing, convergence to a local maximum of the objective function J_ℓ can be achieved for weakly nonlinear systems if the ensemble variance is taken to be small enough. If the ensemble with large spread is used, the estimate can be biased due to the nonlinear terms in Eq. (43). Hence an ensemble with small spread would provide a satisfactory result for weakly nonlinear systems where we can assume the Hessian of J_ℓ is negative definite over the region of interest. However, this iterative algorithm does not necessarily guarantee convergence to the global maximum. If there are multiple peaks in J_ℓ, it might be effectual to start with an ensemble with large spread to approach the global maximum. An ensemble with large spread would grasp a large-scale structure of the objective function because the ensemble approximation of the Jacobian gives the gradient averaged over the region where the ensemble members are distributed under a certain assumption (Raanes et al., 2019). Even if the spread is taken to be large at first, convergence would eventually be achieved by reducing the ensemble spread in each iteration as described in the previous section.

Our formulation refers to the result of a simulation run initialized at the $(m-1)$-th estimate $\bar{\boldsymbol{x}}_{0,m-1}$ for obtaining the m-th estimate in Eq. (31). On the other hand, in many studies, the ensemble mean of simulation runs $\{\boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(i)})\}$ is used as a substitute for $\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})$. If the ensemble mean $\overline{\boldsymbol{g}(\boldsymbol{x}_{0,m-1}^{(i)})}$ is used, the ensemble in each iteration must be generated so as to satisfy Eq. (26), which is not required in our formulation; that is, the ensemble mean must be equal to the $(m-1)$-th estimate $\bar{\boldsymbol{x}}_{0,m-1}$. It should also be kept in mind that some bias due to the nonlinear terms in Eq. (42) could be introduced when the ensemble mean is used instead of $\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})$. However, this bias could be suppressed by taking the ensemble spread to be small. Since the use of the ensemble mean would save the computational cost of one simulation run for each iteration, it might be a useful treatment for practical applications.

It is also important to appropriately choose the parameter σ_m^2. A sufficiently large σ_m^2 guarantees that the m-th estimate $\bar{\boldsymbol{x}}_{0,m}$ is better than the previous estimate $\bar{\boldsymbol{x}}_{0,m-1}$ and hence convergence is stable. However, convergence speed will be degraded

with large σ_m^2 because $\bar{\boldsymbol{x}}_{0,m}$ is strongly constrained by the penalty weighted with σ_m^2. Although there is no definitive way to determine this parameter, $\sigma_m^2/2$ should have a similar scale to the right-hand side of Eq. (50); that is, if the third- and higher-order terms are assumed to be negligible,

$$\sigma_m^2 \sim \mathsf{X}_{m-1}^T \left[\left(\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right)^T \mathsf{R}^{-1} \left(\nabla^2 \boldsymbol{g} \right) \right] \mathsf{X}_{m-1}. \tag{55}$$

Since the right-hand side of Eq. (55) contains $\left(\nabla^2 \boldsymbol{g} \right)$ which comes from a nonlinear term of the function $\boldsymbol{g}$, σ_m^2 should be taken larger as system nonlinearity is severer. This equation also suggests that σ_m^2 should depend on the discrepancy between observation $\boldsymbol{y}$ and the $(m-1)$-th prediction $\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})$. Although $\left(\nabla^2 \boldsymbol{g} \right)$ is unknown in general, $\| \boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \|$ could be used as a guide for determine σ_m^2. The parameter σ_m^2 should also be dependent on the variance of the ensemble. If an ensemble with a large spread is used, σ_m^2 should be set large accordingly.

6 Bayesian form

The algorithm in Section 4 maximizes the log-likelihood function in Eq. (23). However, it would be sometimes required to incorporate prior information into the estimate in a Bayesian manner. We thus consider the following log-posterior density function as the objective function:

$$J(\boldsymbol{x}_0) = -\frac{1}{2} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b} \right)^T \mathsf{P}_{0,b}^{-1} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b} \right) - \frac{1}{2} \left[\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}_0) \right]^T \mathsf{R}^{-1} \left[\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}_0) \right], \tag{56}$$

which is the same as Eq. (10) although the vectors of the whole time sequence are combined into a single vector for each $\boldsymbol{y}$ and $\boldsymbol{g}(\boldsymbol{x}_0)$ as in Eq. (23). Eq. (56) is proportional to the log-posterior distribution when $p(\boldsymbol{x}_0)$ and $p(\boldsymbol{y}|\boldsymbol{x}_0)$ are assumed to be Gaussian. The Taylor expansion of Eq. (56) is

$$\begin{aligned} J(\boldsymbol{x}_0) = & -\frac{1}{2} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b} \right)^T \mathsf{P}_{0,b}^{-1} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,b} \right) - \frac{1}{2} \left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right]^T \mathsf{R}^{-1} \left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right] \\ & + \left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right]^T \mathsf{R}^{-1} \mathsf{G}_{m-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,m-1}) \\ & - \frac{1}{2} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,m-1})^T \mathsf{G}_{m-1}^T \mathsf{R}^{-1} \mathsf{G}_{m-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,m-1}) \\ & + \frac{1}{4} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,m-1})^T \left[\left(\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right)^T \mathsf{R}^{-1} \left(\nabla^2 \boldsymbol{g} \right) \right] (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,m-1}) \\ & + O(\| \boldsymbol{x}_0 - \bar{\boldsymbol{x}}_{0,m-1} \|^3). \end{aligned} \tag{57}$$

Applying Eqs. (38) and (44), we obtain the following approximate objective function:

$$\begin{aligned} J_{w_m}(\boldsymbol{w}_m) = & -\frac{1}{2} \left(\bar{\boldsymbol{x}}_{0,m-1} - \bar{\boldsymbol{x}}_{0,b} + \mathsf{X}_{m-1} \boldsymbol{w}_m \right)^T \mathsf{P}_{0,b}^{-1} \left(\bar{\boldsymbol{x}}_{0,m-1} - \bar{\boldsymbol{x}}_{0,b} + \mathsf{X}_{m-1} \boldsymbol{w}_m \right) \\ & - \frac{1}{2} \left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right]^T \mathsf{R}^{-1} \left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right] \\ & + \left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right]^T \mathsf{R}^{-1} \Gamma_{m-1} \boldsymbol{w}_m - \frac{1}{2} \boldsymbol{w}_m^T \Gamma_{m-1}^T \mathsf{R}^{-1} \Gamma_{m-1} \boldsymbol{w}_m \\ & + \frac{1}{4} \boldsymbol{w}_m^T \mathsf{X}_{m-1}^T \left[\left(\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1}) \right)^T \mathsf{R}^{-1} \left(\nabla^2 \boldsymbol{g} \right) \right] \mathsf{X}_{m-1} \boldsymbol{w}_m + O(\| \boldsymbol{w}_m \|^3). \end{aligned} \tag{58}$$

As per Eq. (50), we can take σ_m^2 so that the fifth and sixth terms on the right-hand side of Eq. (58) can be minorized by a quadratic function $-(\sigma_m^2/2)\boldsymbol{w}_m^T\boldsymbol{w}_m$, and we obtain the following surrogate function which minorizes the function J_{w_m}:

$$\begin{aligned}
J_{w_m}^{\dagger}(\boldsymbol{w}_m|\bar{\boldsymbol{x}}_{0,m-1}) = &-\frac{\sigma_m^2}{2}\boldsymbol{w}_m^T\boldsymbol{w}_m \\
&-\frac{1}{2}\left(\bar{\boldsymbol{x}}_{0,m-1}-\bar{\boldsymbol{x}}_{0,b}+\mathsf{X}_{m-1}\boldsymbol{w}_m\right)^T\mathsf{P}_{0,b}^{-1}\left(\bar{\boldsymbol{x}}_{0,m-1}-\bar{\boldsymbol{x}}_{0,b}+\mathsf{X}_{m-1}\boldsymbol{w}_m\right) \\
&-\frac{1}{2}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T\mathsf{R}^{-1}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right] \\
&+\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T\mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m-\frac{1}{2}\boldsymbol{w}_m^T\mathsf{\Gamma}_{m-1}^T\mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m \\
= &-\frac{\sigma_m^2}{2}\boldsymbol{w}_m^T\boldsymbol{w}_m-\frac{1}{2}(\bar{\boldsymbol{x}}_{0,m-1}-\bar{\boldsymbol{x}}_{0,b})^T\mathsf{P}_{0,b}^{-1}(\bar{\boldsymbol{x}}_{0,m-1}-\bar{\boldsymbol{x}}_{0,b}) \\
&-\frac{1}{2}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T\mathsf{R}^{-1}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right] \\
&-(\bar{\boldsymbol{x}}_{0,m-1}-\bar{\boldsymbol{x}}_{0,b})^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_{m-1}\boldsymbol{w}_m+\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]^T\mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m \\
&-\frac{1}{2}\boldsymbol{w}_m^T\mathsf{X}_{m-1}^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_{m-1}\boldsymbol{w}_m-\frac{1}{2}\boldsymbol{w}_m^T\mathsf{\Gamma}_{m-1}^T\mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\boldsymbol{w}_m, \qquad (59)
\end{aligned}$$

which satisfies the conditions

$$J_{w_m}^{\dagger}(\boldsymbol{w}_m|\bar{\boldsymbol{x}}_{0,m-1}) \le J_{w_m}(\boldsymbol{w}_m) \approx J(\bar{\boldsymbol{x}}_{0,m-1}+\mathsf{X}_{m-1}\boldsymbol{w}_m), \qquad (60)$$

$$J_{w_m}^{\dagger}(\boldsymbol{0}|\bar{\boldsymbol{x}}_{0,m-1}) = J_{w_m}(\boldsymbol{0}) = J(\bar{\boldsymbol{x}}_{0,m-1}). \qquad (61)$$

This function is maximized when

$$\begin{aligned}
\hat{\boldsymbol{w}}_m = &\left(\sigma_m^2\mathbf{I}+\mathsf{X}_{m-1}^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_{m-1}+\mathsf{\Gamma}_{m-1}^T\mathsf{R}^{-1}\mathsf{\Gamma}_{m-1}\right)^{-1} \\
&\times\left(\mathsf{\Gamma}_{m-1}^T\mathsf{R}^{-1}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]-\mathsf{X}_{m-1}^T\mathsf{P}_{0,b}^{-1}(\bar{\boldsymbol{x}}_{0,m-1}-\bar{\boldsymbol{x}}_{0,b})\right). \qquad (62)
\end{aligned}$$

The m-th estimate for $\boldsymbol{x}_0$ is obtained as

$$\bar{\boldsymbol{x}}_{0,m} = \bar{\boldsymbol{x}}_{0,m-1}+\mathsf{X}_{m-1}\hat{\boldsymbol{w}}_m. \qquad (63)$$

Similarly to Eq. (54), we obtain

$$J(\bar{\boldsymbol{x}}_{0,m-1}) = J_{w_m}(\boldsymbol{0}) \le J_{w_m}(\hat{\boldsymbol{w}}_m) \approx J(\bar{\boldsymbol{x}}_{0,m-1}+\mathsf{X}_{m-1}\hat{\boldsymbol{w}}_m) = J_{\ell}(\bar{\boldsymbol{x}}_{0,m}). \qquad (64)$$

Thus, $\bar{\boldsymbol{x}}_{0,m}$ is a better estimate than $\bar{\boldsymbol{x}}_{0,m-1}$ if the approximation in Eq. (44) is valid. Generating the $(m+1)$-th ensemble around $\bar{\boldsymbol{x}}_{0,m}$, we can obtain the $(m+1)$-th surrogate function according to Eq. (59) and proceed to the next iteration.

There are various methods for updating the ensemble including the methods mentioned in Section 4. Eq. (32) or (34) is convenient for practical problems because we can avoid computing the inverse of $\mathsf{P}_{0,b}$ in Eq. (62). When Eq. (32) is used for updating the ensemble, we can easily avoid computing the inverse of $\mathsf{P}_{0,b}$ by drawing initial ensemble members from the prior distribution $\mathcal{N}(\boldsymbol{x}_0;\boldsymbol{x}_{0,b},\mathsf{P}_{0,b})$. If initial ensemble members $\{\boldsymbol{x}_{0,0}^{(1)},\ldots,\boldsymbol{x}_{0,0}^{(N)}\}$ obey the prior distribution, we can use the same approximation as Eq. (13); that is,

$$\mathsf{X}_{0,b}^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_{0,b} \approx \mathbf{I}. \qquad (65)$$

Applying Eq. (63) recursively, $\bar{\boldsymbol{x}}_{0,m-1}$ can be reduced to

$$\begin{aligned}\bar{\boldsymbol{x}}_{0,m-1} &= \bar{\boldsymbol{x}}_{0,m-2} + \mathsf{X}_{m-2}\hat{\boldsymbol{w}}_{m-1} = \bar{\boldsymbol{x}}_{0,m-3} + \mathsf{X}_{m-3}\hat{\boldsymbol{w}}_{m-2} + \mathsf{X}_{m-2}\hat{\boldsymbol{w}}_{m-1} \\ &= \cdots = \bar{\boldsymbol{x}}_{0,0} + \sum_{i=1}^{m-1} \mathsf{X}_{i-1}\hat{\boldsymbol{w}}_i \\ &= \bar{\boldsymbol{x}}_{0,0} + \mathsf{X}_{0,b} \sum_{i=1}^{m-1} \alpha_{i-1}\hat{\boldsymbol{w}}_i \end{aligned} \tag{66}$$

Inserting Eqs. (32) and (66) into Eq. (62) and applying Eq. (65), we obtain

$$\hat{\boldsymbol{w}}_m \approx \left([\sigma_m^2 + \alpha_{m-1}^2]\mathbf{I} + \Gamma_{m-1}^T \mathsf{R}^{-1} \Gamma_{m-1}\right)^{-1} \left(\Gamma_{m-1}^T \mathsf{R}^{-1} \left[\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right] - \alpha_{m-1} \sum_{i=1}^{m-1} \alpha_{i-1}\hat{\boldsymbol{w}}_i \right). \tag{67}$$

Thus, we can avoid computing the inverse of $\mathsf{P}_{0,b}$. Likewise, when Eq. (34) is used for updating the ensemble, we can apply Eq. (65) to avoid computing the inverse of $\mathsf{P}_{0,b}$ (See Appendix).

As described in the previous section, the use of Eq. (32) of (34) confines the estimate $\bar{\boldsymbol{x}}_{0,m-1}$ within the subspace spanned by the initial ensemble. On the other hand, Eq. (36) enables us to seek the optimal value of $\boldsymbol{x}_0$ in a different subspace in each iteration. We can then obtain the local maximum in the full vector space if Q_m is taken to be full rank. It appears that a similar approximation to Eq. (67) is applicable if Q_m is taken to be a scalar matrix of $\mathsf{P}_{0,b}$. However, since this approximation considers a different approximate objective function in each iteration, monotonic convergence is not guaranteed. In order to ensure monotonic convergence, Eq. (36) requires the computation of the inverse of $\mathsf{P}_{0,b}$ in general. Nonetheless, if $\mathsf{P}_{0,b}^{-1}$ can be obtained, the method with Eq. (36) would be helpful for improving the estimate.

7 Experiments

Preceding studies have already demonstrated the usefulness of the ensemble-based iterative algorithms for various data assimilation problems. Estimation with the ensemble update in Eq. (32) has been verified in detail (e.g., Bocquet and Sakov, 2014). The iterative algorithm ensemble update in Eq. (34) has also been demonstrated (e.g., Minami et al., 2020). Although it might not be necessary to show the ability of the ensemble-based iterative algorithm further, we here verify some properties suggested in the above discussion through twin experiments with a simple model rather than a practical model.

In this section, we employ the Lorenz 96 model (Lorenz and Emanuel, 1998), which is written by the following equations:

$$\frac{dx_m}{dt} = (x_{m+1} - x_{m-2})x_{m-1} - x_m + f \tag{68}$$

for $m = 1, \ldots, M$, where $x_{-1} = x_{M-1}$, $x_0 = x_M$, and $x_{M+1} = x_1$. The state dimension M was taken to be 40 and the forcing term f was taken to be 8. The true scenario was generated by running the model with a certain initial state. We here consider a weakly nonlinear problem. The assimilation window was accordingly taken as a short time interval $0 < t \leq 8$. It was assumed that all the state variables could be observed with a fixed time interval ($\Delta t = 0.1$), and hence, 80 data were generated for each state variable. The observation noise for each variable was assumed to independently follow a Gaussian distribution with

mean 0 and standard deviation 0.5. In each data assimilation experiment, the prior distribution was assumed to be a Gaussian distribution with mean $\mathbf{0}$ and variance $\zeta^2\mathbf{I}$, $\mathcal{N}(\mathbf{0}, \zeta^2\mathbf{I})$, where $\zeta = 5$.

We compare two ensemble updating methods of Eqs. (32) and (36). In applying Eq. (32), the initial ensemble $\{\boldsymbol{x}_{0,0}^{(1)}, \ldots, \boldsymbol{x}_{0,0}^{(N)}\}$ was drawn from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \varepsilon^2\mathbf{I})$ where $\varepsilon = 5 \times 10^{-6}$ and X_0 was obtained as follows:

$$\mathsf{X}_0 = \frac{1}{\sqrt{N}} \left(\boldsymbol{x}_{0,0}^{(1)} - \bar{\boldsymbol{x}}_{0,m} \quad \cdots \quad \boldsymbol{x}_{0,0}^{(N)} - \bar{\boldsymbol{x}}_{0,m} \right). \tag{69}$$

The matrix X_m for each iteration was fixed at $\mathsf{X}_m = \mathsf{X}_0$, which corresponds to the setting in Eq. (32) with $\alpha_m = 1$. The discussion in Section 5 suggests that the penalty parameter σ_m^2 should be determined according to Eq. (55). Although $\left(\nabla^2 \boldsymbol{g}\right)$ is unknown, we can say that σ_m^2 should be related with the variance of the ensemble and the discrepancy between $\boldsymbol{y}$ and $\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})$. We thus gave σ_m^2 as follows:

$$\sigma_m^2 = \delta^2 \sqrt{(\boldsymbol{y}_K - \boldsymbol{g}_K(\hat{\boldsymbol{x}}_{0,m-1}))^T \mathsf{R}_K^{-1} (\boldsymbol{y}_K - \boldsymbol{g}_K(\hat{\boldsymbol{x}}_{0,m-1}))} \mathrm{tr}\left(\mathsf{\Gamma}_{m-1}^T \mathsf{R}^{-1} \mathsf{\Gamma}_{m-1}\right), \tag{70}$$

where we tried two cases with $\delta = 1.5 \times 10^{-3}$ and $\delta = 1.5 \times 10^{-2}$. Here the part of the square root of a quadratic form of $\boldsymbol{y}_K - \boldsymbol{g}_K(\hat{\boldsymbol{x}}_{0,m-1})$ was multiplied in order that σ_m^2 was roughly proportional to $\|\boldsymbol{y} - \boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\|$, and $\mathrm{tr}\left(\mathsf{\Gamma}_{m-1}^T \mathsf{R}^{-1} \mathsf{\Gamma}_{m-1}\right)$ was for representing the variance of the ensemble.

Figures 1 and 2 shows results with Eq. (32) where $\delta = 1.5 \times 10^{-3}$ and $\delta = 1.5 \times 10^{-2}$, respectively. We took the ensemble size N to be 30, which is less than the state dimension, and performed the estimation 20 times with different seeds of a pseudo random number generator. The value of the objective function J in Eq. (56) for each iteration is plotted for each of 20 trials in these figures. When $\delta = 1.5 \times 10^{-3}$, the value of J tended to increase more sharply than when $\delta = 1.5 \times 10^{-2}$. However, J did not monotonically increase when $\delta = 1.5 \times 10^{-3}$, while it monotonically increased when $\delta = 1.5 \times 10^{-2}$. According to the discussion in Section 5, monotonic convergence is achieved when σ_m^2 is taken to be large enough. However, convergence speed becomes slow when σ_m^2 is large. The results in Figures 1 and 2 thus confirmed our discussion on the convergence. However, the results shown in Figures 1 and 2 did not converge to the same value, which means the results depended on the seeds of pseudo random numbers. This would indicate that a local maximum within a subspace spanned by the ensemble does not match the maximum in the full state vector space and that the value of the local maximum depends on the subspace.

Figures 3 and 4 shows results with Eq. (36) where $\delta = 1.5 \times 10^{-3}$ and $\delta = 1.5 \times 10^{-2}$, respectively. Again, the ensemble size N was taken to be 30, and the results of 20 trials with different seeds of pseudo random numbers are overplotted. Again, when $\delta = 1.5 \times 10^{-3}$, the increase of J tended to be sharp while it was not monotonic. On the other hand, when $\delta = 1.5 \times 10^{-2}$, the increase of J was gradual but monotonic. In contrast with the results in Figures 3 and 4, the values of J in different trials converged to the same value after about 15 iterations in the case with $\delta = 1.5 \times 10^{-3}$ shown in Figure 3. In the case with $\delta = 1.5 \times 10^{-2}$, the convergence was much slower, but the values of J converged to the same value as the the case with $\delta = 1.5 \times 10^{-2}$ after about 80 iterations in all of the 20 trials (not shown). These results shows that the maximum of the objective function in the full vector space can be reached by changing an ensemble in each iteration even if the ensemble does not span the full vector space.

In order to closely investigate the effect of σ_m^2, we conducted additional experiments for a case in which nonlinearity is a little stronger. While Figures 3 and 4 show the results when the assimilation window was taken as $0 < t \leq 8$, Figures 5 and 6

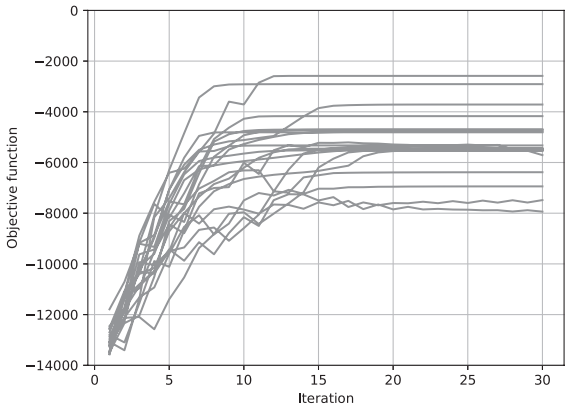

Figure 1. The value of the objective function J for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (32) with $\delta = 1.5 \times 10^{-3}$.

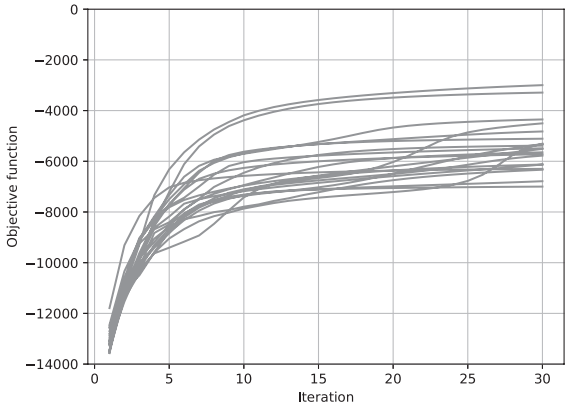

Figure 2. The value of the objective function J for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (32) with $\delta = 1.5 \times 10^{-2}$.

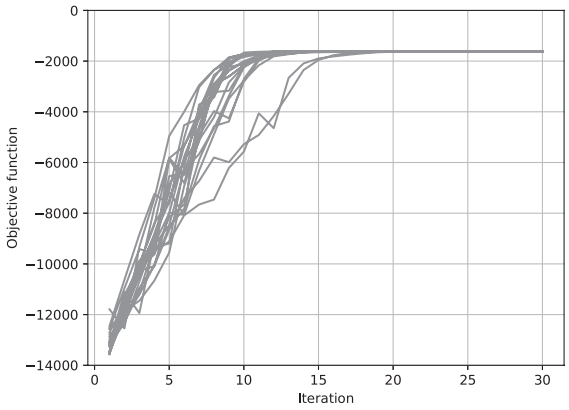

Figure 3. The value of the objective function J for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (36) with $\delta = 1.5 \times 10^{-3}$.

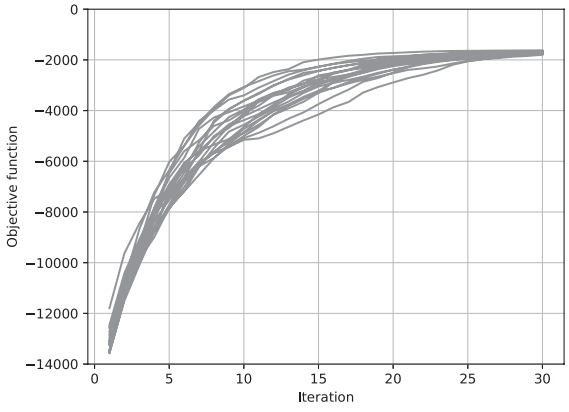

Figure 4. The value of the objective function J for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (36) with $\delta = 1.5 \times 10^{-2}$.

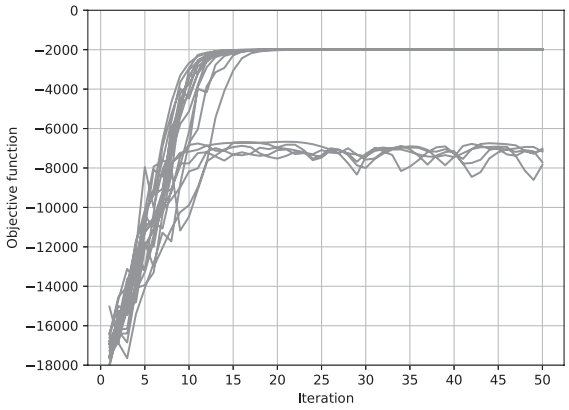

Figure 5. The value of the objective function J for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (36) with $\delta = 1.5 \times 10^{-3}$ and the ensemble window was taken as $0 < t \leq 10$.

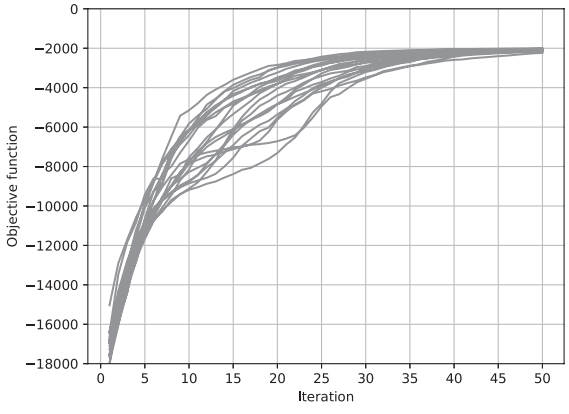

Figure 6. The value of the objective function J for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (36) with $\delta = 1.5 \times 10^{-2}$ and the ensemble window was taken as $0 < t \leq 10$.

show the results with a little longer assimilation window, $0 < t \leq 10$. Although the other settings were the same as Figures 3 and 4, the effect of the nonlinearity on the objective function J was a little severer due to the longer assimilation window. When $\delta = 1.5 \times 10^{-3}$, the J value converged to about -2000 in many of the 20 trials. In some trials, however, J did not converge but oscillated below -6000. In contrast, when δ was as large as 1.5×10^{-2}, the J value converged to the same value after about 50 iterations in all of the 20 trials. As discussed in Section 5, a sufficiently large σ_m^2 guarantees that the estimate is improved in each iteration. Although convergence speed becomes worse, stable estimation can be attained.

We also conducted experiments with a higher-dimensional system. The method with a randomly generated ensemble was applied to the Lorenz 96 model with 400 variables ($M = 400$), of which the dimension is ten times higher. Figure 7 shows the result with 400 variables. The assimilation was taken as $0 < t \leq 8$ and δ was set at 1.5×10^{-3}, which are the same as in Figures 3. The ensemble size N was taken to be 200. For the assimilation into the Lorenz 96 model with 400 variables, the convergence was attained in about 20 iterations with 200 ensemble members. In this high-dimensional case, monotonic convergence was achieved even if δ was taken to be as small as in Figure 3. As far as we conducted experiments with the Lorenz 96 model with various dimensions, the convergence becomes stabler as the state dimension M becomes higher. This might imply that the nonlinear term (the fifth term of the right-hand side of Eq. (58)) is depressed in the high-dimensional Lorenz 96 models. However, we have not resolved the reason of the stable convergence for the high-dimensional Lorenz 96 systems at present.

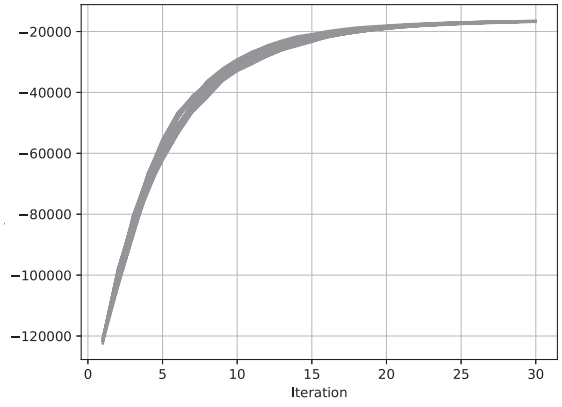

Figure 7. The value of the objective function J for each iteration for 20 trials of the estimation for the 400 dimensional system. The ensemble was updated using Eq. (36) with $\delta = 1.5 \times 10^{-3}$.

8 Discussion and conclusions

The ensemble variational method is derived under the assumption that a linear approximation of a dynamical system model is valid over a range of uncertainty. This linear approximation is not valid in such problems that the scale of uncertainty is much larger than the range of linearity. However, a local maximum of the log-likelihood or log-posterior function can be attained by updating the ensemble iteratively even in cases with a large uncertainty. The present paper assessed the influence of system nonlinearity on this iterative algorithm after considering the nonlinear terms of the system function $\boldsymbol{g}$. The discussion suggests two points to guarantee the monotonic convergence to a local maximum in the subspace spanned by the ensemble. One is that the ensemble spread must be set to be small, and the other is that the penalty parameter σ_m^2 must be set to be large enough. A sufficiently large σ_m^2 would ensure monotonic convergence, although convergence speed would become poorer with a too large

σ_m^2. The effect of this penalty term has also been experimentally confirmed in Section 7. These properties would be reasonable if this iterative algorithm is regarded as an approximation of the Levenberg-Marquardt method.

In applying the iterative algorithm discussed in this paper, the choice of the parameter σ_m^2 would be an important issue. Although it was determined according to Eq. (70) in Section 7, Eq. (70) still requires to tune the parameter δ. However, it is not necessary to finely tune δ because δ would not make a crucial effect on the performance of the algorithm. It would thus be enough to roughly determine δ. In addition, one could check whether the objective function J increases or not at each iteration just by running one forward simulation initialized at $\bar{\boldsymbol{x}}_{0,m}$. In the iterative algorithm, most computational cost is spent for running the ensemble simulation with multiple initial states. The pilot run, which is computationally much cheaper than the ensemble run, would be a feasible way for tuning δ in practical cases.

One issue peculiar to the ensemble-based method is the rank deficiency which occurs when the ensemble size is smaller than the dimension of the initial state $\boldsymbol{x}_0$. If the ensemble is confined within a particular subspace, the iterative algorithm can only attain the optimal value within the subspace spanned by the ensemble. However, our discussion indicates that, if σ_m^2 is sufficiently large, it is ensured that the discrepancies between estimates and observations are reduced in each iteration even if the ensemble is confined within a subspace. If the ensemble is updated so as to span a different subspace in each iteration, the optimal solution would be sought in a different subspace in each iteration, and the estimate would converge to a local minimum after infinite iterations.

Comparing with the adjoint method, which is a conventional variational method for 4-dimensional variational problems, the convergence rate of this iterative method would be poorer because it employs an ensemble approximation within a lower-dimensional subspace at each iteration. Nonetheless, we can say that the iterative ensemble-based method is potentially useful because it is much easier to implement. While the adjoint method requires an adjoint code which is usually time-consuming to develop, the ensemble-based method can solve the same problem without requiring an adjoint code. This paper mainly considers data assimilation problems. However, the framework of the iterative ensemble variational method is also applicable to general nonlinear inverse problems as far as the Gaussian assumption in Eq. (23) or Eq. (56) is upheld. If an ensemble of the results of forward runs is available, many practical problems can readily be addressed. This method could therefore be a promising tool for data assimilation and various inverse problems.

Code availability. The code for reproducing the experimental results shown in Section 7 is available at the following Web site.
`http://daweb.ism.ac.jp/~shiny/codes/npg2020.zip`

Appendix A: Algorithm for Bayesian estimation with ensemble transform

In the following, it is described how the iteration can be performed without computing the inverse of $\mathsf{P}_{0,b}$ when the ensemble is updated with the ensemble transform scheme in Eq. (34). When the ensemble is updated by the ensemble transform in the

manner of Eq. (34):

$$\mathsf{X}_m = \mathsf{X}_{m-1}\mathsf{T}_m, \tag{A1}$$

the transform matrix T_m should be given as

$$\mathsf{T}_m = \mathsf{U}_m(\mathbf{I}+\Lambda_m)^{-\frac{1}{2}}\mathsf{U}_m^T, \tag{A2}$$

where U_m and Λ_m are obtained by the following eigenvalue decomposition:

$$\mathsf{U}_m\Lambda_m\mathsf{U}_m^T = \sigma_m^{-2}(\mathsf{X}_{m-1}^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_{m-1} + \Gamma_{m-1}^T\mathsf{R}^{-1}\Gamma_{m-1}). \tag{A3}$$

If X_{m-1} is obtained according to Eq. (A1),

$$\mathsf{X}_{m-1} = \mathsf{X}_{m-2}\mathsf{T}_{m-1} = \mathsf{X}_0\mathsf{T}_1\mathsf{T}_2\cdots\mathsf{T}_{m-1}. \tag{A4}$$

Defining the matrix C_{m-1} as

$$\mathsf{C}_{m-1} = \mathsf{T}_1\mathsf{T}_2\cdots\mathsf{T}_{m-1}, \tag{A5}$$

X_{m-1} can be written as

$$\mathsf{X}_{m-1} = \mathsf{X}_0\mathsf{C}_{m-1}. \tag{A6}$$

If the initial ensemble is sampled from the prior distribution $\mathcal{N}(\boldsymbol{x}_0;\bar{\boldsymbol{x}}_{0,b},\mathsf{P}_{0,b})$, we can apply Eq. (65) again. Using Eqs. (65) and (A6), the term $\mathsf{X}_{m-1}^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_{m-1}$ in Eq. (62) can be reduced to

$$\mathsf{X}_{m-1}^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_{m-1} = \mathsf{C}_{m-1}^T\mathsf{C}_{m-1}. \tag{A7}$$

The m-th estimate is broken down as follows:

$$\begin{aligned}\bar{\boldsymbol{x}}_{0,m-1} &= \bar{\boldsymbol{x}}_{0,m-2} + \mathsf{X}_{m-2}\hat{\boldsymbol{w}}_{m-1} = \bar{\boldsymbol{x}}_{0,b} + \sum_{i=1}^{m-1}\mathsf{X}_{i-1}\hat{\boldsymbol{w}}_i \\ &= \bar{\boldsymbol{x}}_{0,b} + \mathsf{X}_0\sum_{i=1}^{m-1}\mathsf{C}_{i-1}\hat{\boldsymbol{w}}_i,\end{aligned} \tag{A8}$$

where $\mathsf{C}_0 = \mathbf{I}$. Defining a vector $\boldsymbol{\xi}_{m-1}$ as

$$\boldsymbol{\xi}_{m-1} = \sum_{i=1}^{m-1}\mathsf{C}_{i-1}\hat{\boldsymbol{w}}_i, \quad (\boldsymbol{\xi}_0 = \mathbf{0}), \tag{A9}$$

Eq. (A8) becomes

$$\bar{\boldsymbol{x}}_{0,m-1} = \bar{\boldsymbol{x}}_{0,b} + \mathsf{X}_0\boldsymbol{\xi}_{m-1}, \tag{A10}$$

and we obtain

$$\begin{aligned}\mathsf{X}_{m-1}^T\mathsf{P}_{0,b}^{-1}(\bar{\boldsymbol{x}}_{0,m-1}-\bar{\boldsymbol{x}}_{0,b}) &= \mathsf{C}_{m-1}^T\mathsf{X}_0^T\mathsf{P}_{0,b}^{-1}\mathsf{X}_0\boldsymbol{\xi}_{m-1}\\ &\approx \mathsf{C}_{m-1}^T\boldsymbol{\xi}_{m-1}. \end{aligned} \tag{A11}$$

Using Eqs. (65) and (A11), we can rewrite Eq. (62) into a form without the inverse of $\mathsf{P}_{0,b}$:

$$\hat{\boldsymbol{w}}_m = \left(\sigma_m^2\mathbf{I}+\mathsf{C}_{m-1}^T\mathsf{C}_{m-1}+\Gamma_{m-1}^T\mathsf{R}^{-1}\Gamma_{m-1}\right)^{-1}\left(\Gamma_{m-1}^T\mathsf{R}^{-1}\left[\boldsymbol{y}-\boldsymbol{g}(\bar{\boldsymbol{x}}_{0,m-1})\right]-\mathsf{C}_{m-1}^T\boldsymbol{\xi}_{m-1}\right). \tag{A12}$$

The algorithm with the ensemble transform is summarized in Algorithm 2.

Algorithm 2 Iterative algorithm for maximizing the Bayesian objective function J with ensemble transform.

Let $m=1$.
Give an initial estimate of $\boldsymbol{x}_0$, $\bar{\boldsymbol{x}}_{0,1}$.
Give an initial square root of the covariance, X_1.
Let $\mathsf{C}_0=\mathbf{I}$ and $\boldsymbol{\xi}_0=\mathbf{0}$.
while unconverged **do**
  Generate an ensemble $\{\boldsymbol{x}_{0,m}\}_{i=1}^N$ around with a mean of $\bar{\boldsymbol{x}}_{0,m}$ and a variance of $\mathsf{X}_m\mathsf{X}_m^T$.
  Obtain $\Gamma_{k,m-1}$ in Eq. (28).
  Compute $\hat{\boldsymbol{w}}_m$ in Eq. (A12).
  Compute the m-th estimate $\bar{\boldsymbol{x}}_{0,m}$ according to Eq. (63).
  Compute the matrix X_m according to Eq. (A1).
  Let $\mathsf{C}_m=\mathsf{C}_{m-1}\mathsf{T}_{m-1}$
  Let $\boldsymbol{\xi}_m=\boldsymbol{\xi}_{m-1}+\mathsf{C}_{m-1}\hat{\boldsymbol{w}}_m$
  Let $m:=m+1$
end while

Author contributions. The entire study has been conducted by SN.

Competing interests. The author declares that he has no competing interests.

Acknowledgements. This work was in part supported by PRC JSPS CNRS, Bilateral Joint Research Project "Forecasting the geomagnetic secular variation based on data assimilation" and JSPS KAKENHI Grant Number 17H01704.

References

Bannister, R. N.: A review of operational methods of variational and ensemble-variational data assimilation, Q. J. R. Meteorol. Soc., 143, 607–633, https://doi.org/10.1002/qj.2982, 2017.

Bishop, C. H., Etherton, B. J., and Majumdar, S. J.: Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects, Mon. Wea. Rev., 129, 420–436, 2001.

Bocquet, M. and Sakov, P.: Combining inflation-free and iterative ensemble Kalman filters for strongly nonlinear systems, Nonlin. Process. Geophys., 19, 383–399, https://doi.org/10.5194/npg-19-383-2012, 2012.

Bocquet, M. and Sakov, P.: Joint state and parameter estimation with iterative ensemble Kalman smoother, Nonlin. Process. Geophys., 20, 803–818, https://doi.org/10.5194/npg-20-803-2013, 2013.

Bocquet, M. and Sakov, P.: An iterative ensemble Kalman smoother, Q. J. R. Meteorol. Soc., 140, 1521–1535, https://doi.org/10.1002/qj.2236, 2014.

Böhning, D. and Lindsay, B. G.: Monotonicity of quadratic-approximation algorithms, Ann. Inst. Statist. Math., 40, 641–663, 1988.

Buehner, M.: Ensemble-derived stationary and flow-dependent background-error covariances: Evaluation in a quasi-operational NWP setting, Q. J. R. Meteorol. Soc., 131, 1013–1043, 2005.

Buehner, M., Houtekamer, P. L., Charette, C., Mitchell, H. L., and He, B.: Intercomparison of variational data assimilation and the ensemble Kalman filter for global deterministic NWP. Part I: Description and single-observation experiment, Mon. Wea. Rev., 138, 1550–1566, 2010.

Chen, Y. and Oliver, D. S.: Ensemble randomized maximum likelihood method as an iterative ensemble smoother, Math. Geosci., 44, 1–26, 2012.

Courtier, P., Thépaut, J.-N., and Hollingsworth, A.: A strategy for operational implementation of 4D-Var, using an incremental approach, Q. J. R. Meteorol. Soc., 120, 1367–1387, 1994.

Emerick, A. A. and Reynolds, A. C.: History matching time-lapse seismic data using the ensemble Kalman filter with multiple data assimilation, Computat. Geosci., 16, 639–659, https://doi.org/10.1007/s10596-012-9275-5, 2012.

Emerick, A. A. and Reynolds, A. C.: Ensemble smoother with multiple data assimilation, Computers Geosci., 55, 3–15, https://doi.org/10.1016/j.cageo.2012.03.011, 2013.

Evensen, G.: Analysis of iterative ensemble smoothers for solving inverse problems, Computat. Geosci., 22, 885–908, https://doi.org/10.1007/s10596-018-9731-y, 2018.

Evensen, G. and van Leeuwen, P. J.: An emsemble Kalman smoother for nonlinear dynamics, Mon. Wea. Rev., 128, 1852–1867, 2000.

Godinez, H. C., Yu, Y., Lawrence, E., Henderson, M. G., Larsen, B., and Jordanova, V. K.: Ring current pressure estimation with RAM-SCB using data assimilation and Van Allen Probe flux data, Geophys. Res. Lett., 43, 11 948–11 956, https://doi.org/10.1002/2016GL071646, 2016.

Gu, Y. and Oliver, D. S.: An iterative ensemble Kalman filter for multiphase fluid flow data assimilation, SPE J., 12, 438–446, 2007.

Kalnay, E. and Yang, S.-C.: Accelerating the spin-up of ensemble Kalman filtering, Q. J. R. Meteorol. Soc., 136, 1644–1651, 2010.

Kano, M., Miyazaki, S., Ishikawa, Y., Hiyoshi, Y., Ito, K., and Hirahara, K.: Real data assimilation for optimization of frictional parameters and prediction of afterslip in the 2003 Tokachi-oki earthquake inferred from slip velocity by an adjoint method, Geophys. J. Int., 203, 646–663, https://doi.org/10.1093/gji/ggv289, 2015.

Lange, K.: MM optimization algorithms, SIAM, Philadelphia, 2016.

Lange, K., Hunter, D. R., and Yang, I.: Optimization transfer using surrogate objective functions, J. Comput. Graph. Statist., 9, 1–20, 2000.

Lawless, A. S., Gratton, S., and Nichols, N. K.: An investigation of incremental 4D-Var using non-tangent linear model, Q. J. R. Meteorol. Soc., 131, 459–476, 2005.

Liu, C., Xiao, Q., and Wang, B.: An ensemble-based four-dimensional variational data assimilation scheme. Part I: Technical formulation and preliminary test, Mon. Wea. Rev., 136, 3363–3373, 2008.

Liu, C., Xiao, Q., and Wang, B.: An ensemble-based four-dimensional variational data assimilation scheme. Part II: Observing system simulation experiments with advanced research WRF (ARW), Mon. Wea. Rev., 137, 1687–1704, 2009.

Livings, D. M., Dance, S. L., and Nichols, N. K.: Unbiased ensemble square root filters, Physica D, 237, 1021–1028, 2008.

Lorenc, A. C.: The potential of the ensemble Kalman filter for NWP–a comparison with 4D-Var, Q. J. R. Meteorol. Soc., 129, 3183–3203, 2003.

Lorenz, E. N. and Emanuel, K. A.: Optimal sites for supplementary weather observations: Simulations with a small model, J. Atmos. Sci., 55, 399, 1998.

Minami, T., Nakano, S., Lesur, V., Takahashi, F., Matsushima, M., Shimizu, H., Nakashima, R., Taniguchi, H., and Toh, H.: A candidate secular variation model for IGRF-13 based on MHD dynamo simulation and 4DEnVar data assimilation, Earth Planets Space, accepted, 2020.

Nakano, S., Ueno, G., Ebihara, Y., Fok, M.-C., Ohtani, S., Brandt, P. C., Mitchell, D. G., Keika, K., and Higuchi, T.: A method for estimating the ring current structure and the electric potential distribution using ENA data assimilation, J. Geophys. Res., 113, A05 208, https://doi.org/10.1029/2006JA011853, 2008.

Nocedal, J. and Wright, S. J.: Numerical optimization, 2nd ed., Springer, New York, 2006.

Raanes, P. N., Stordal, A. S., and Evensen, G.: Revisiting the stochastic iterative ensemble smoother, Nonlin. Process. Geophys., 26, 325–338, https://doi.org/10.5194/npg-26-325-2019, 2019.

Sanchez, S., Wicht, J., Bärenzung, J., and Holschneider, M.: Sequential assimilation of geomagnetic observations: perspectives for the reconstruction and prediction of core dynamics, Geophys. J. Int., 217, 1434–1450, https://doi.org/10.1093/gji/ggz090, 2019.

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

Yokota, S., Kunii, M., Aonashi, K., and Origuchi, S.: Comparison between four-dimensional LETKF and emsemble-based variational data assimilation with observation localization, SOLA, 12, 80–85, https://doi.org/10.2151/sola.2016-019, 2016.

Zupanski, M., Navon, M., and Zupanski, D.: The Maximum likelihood ensemble filter as a non-differentiable minimization algorithm, Q. J. R. Meteorol. Soc., 134, 1039–1050, 2008.