



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

Shin'ya Nakano<sup>1,2</sup>

<sup>1</sup>Department of Statistical Modeling, The Institute of Statistical Mathematics, Tachikawa, Japan

<sup>2</sup>Center for Data Assimilation Research and Applications, Joint Support Center for Data Science Research, Tachikawa, Japan

**Correspondence:** Shin'ya Nakano (shiny@ism.ac.jp)

Received: 5 April 2020 – Discussion started: 17 April 2020

Revised: 29 December 2020 – Accepted: 5 January 2021 – Published: 8 February 2021

**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 4D 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, in which 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 a 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 4D 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, then 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), the timescale of the 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 estimations 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, 4D 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 a 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 Sect. 2, the formulation of the variational data assimilation problem is described. In Sect. 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 Sect. 4, and the behavior of the iterative algorithm is evaluated in Sect. 5. In Sect. 6, a Bayesian extension is introduced. Section 7 experimentally verifies our findings. Finally, a discussion and conclusions are presented in Sect. 8.

## 2 The 4D variational data assimilation (4DEnVar)

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

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}), \quad (1)$$

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

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{w}_k, \quad (2)$$

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

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

The likelihood of  $\mathbf{x}_k$  given  $\mathbf{y}_k$  is as follows:

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

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

$$\mathbf{h}_k(\mathbf{x}_k) = \mathbf{g}_k(\mathbf{x}_0), \quad (5)$$

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

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

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

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

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

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

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

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

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

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

### 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  $f_k$  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  $\{\mathbf{x}_{0:K}^{(1)}, \dots, \mathbf{x}_{0:K}^{(N)}\}$ , where  $\mathbf{x}_{0:K}$  indicates the whole sequence of the states from  $t_0$  to  $t_K$ ; that is,  $\mathbf{x}_{0:K} = (\mathbf{x}_0^T \cdots \mathbf{x}_K^T)^T$ . The initial state of each ensemble member  $\mathbf{x}_0^{(i)}$  is assumed to be sampled from the Gaussian distribution  $\mathcal{N}(\mathbf{x}_0; \mathbf{x}_{0,b}, \mathbf{P}_{0,b})$ . The objective function in Eq. (10) is approximated by using this ensemble.

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

$$\mathbf{X}_{0,b} = \frac{1}{\sqrt{N}} \begin{pmatrix} \mathbf{x}_0^{(1)} - \bar{\mathbf{x}}_{0,b} & \cdots & \mathbf{x}_0^{(N)} - \bar{\mathbf{x}}_{0,b} \end{pmatrix}. \quad (11)$$

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

$$\mathbf{x}_0 = \bar{\mathbf{x}}_{0,b} + \mathbf{X}_{0,b} \mathbf{w}. \quad (12)$$

This assumption means that  $\mathbf{x}_0$  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  $\mathbf{X}_{0,b}$  is  $N (< \dim \mathbf{x}_0)$  and approximate the inverse of  $\mathbf{P}_{0,b}$  by the Moore–Penrose inverse matrix of  $\mathbf{X}_{0,b} \mathbf{X}_{0,b}^T$ , the first term of the right-hand side of Eq. (10) can be approximated as follows:

$$\begin{aligned} & -\frac{1}{2}(\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b})^T \mathbf{P}_{0,b}^{-1}(\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}) \\ & = -\frac{1}{2} \mathbf{w}^T \mathbf{X}_{0,b}^T \mathbf{P}_{0,b}^{-1} \mathbf{X}_{0,b} \mathbf{w} \approx -\frac{1}{2} \mathbf{w}^T \mathbf{w}. \end{aligned} \quad (13)$$

This corresponds to a low-rank approximation within the subspace spanned by the ensemble members. The prior mean  $\bar{\mathbf{x}}_{0,b}$  is usually given by the ensemble mean of  $\{\mathbf{x}_0^{(i)}\}_{i=1}^N$ . 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  $g_k(\mathbf{x}_0)$  is approximated based on the first-order Taylor expansion as follows:

$$\begin{aligned} g_k(\mathbf{x}_0) & \approx g_k(\bar{\mathbf{x}}_{0,b}) + \mathbf{G}_k(\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}) \\ & \approx g_k(\bar{\mathbf{x}}_{0,b}) + \mathbf{G}_k \mathbf{X}_{0,b} \mathbf{w}, \end{aligned} \quad (14)$$

where  $\mathbf{G}_k$  is the Jacobian of  $g_k$  at  $\bar{\mathbf{x}}_{0,b}$ . The matrix  $\mathbf{G}_k \mathbf{X}_{0,b}$  in Eq. (14) is approximated as follows:

$$\begin{aligned} \mathbf{G}_k \mathbf{X}_{0,b} & \approx \frac{1}{\sqrt{N}} \\ & \times \begin{pmatrix} g_k(\mathbf{x}_0^{(1)}) - g_k(\bar{\mathbf{x}}_{0,b}) & \cdots & g_k(\mathbf{x}_0^{(N)}) - g_k(\bar{\mathbf{x}}_{0,b}) \end{pmatrix}. \end{aligned} \quad (15)$$

Defining the right-hand side of Eq. (15) as  $\mathbf{\Gamma}_k$ , that is, in the following:

$$\begin{aligned} \mathbf{\Gamma}_k & = \frac{1}{\sqrt{N}} \\ & \times \begin{pmatrix} g_k(\mathbf{x}_0^{(1)}) - g_k(\bar{\mathbf{x}}_{0,b}) & \cdots & g_k(\mathbf{x}_0^{(N)}) - g_k(\bar{\mathbf{x}}_{0,b}) \end{pmatrix} \\ & \approx \mathbf{G}_k \mathbf{X}_{0,b}, \end{aligned} \quad (16)$$

we obtain a further approximation of the function  $g_k(\mathbf{x}_0)$  in Eq. (14) as follows:

$$g_k(\mathbf{x}_0) \approx g_k(\bar{\mathbf{x}}_{0,b}) + \mathbf{\Gamma}_k \mathbf{w} \quad (17)$$

(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  $\mathbf{w}$  as follows:

$$\begin{aligned} \hat{J}_w(\mathbf{w}) & = -\frac{1}{2} \mathbf{w}^T \mathbf{w} - \frac{1}{2} \sum_{k=1}^K (\mathbf{y}_k - g_k(\bar{\mathbf{x}}_{0,b}) - \mathbf{\Gamma}_k \mathbf{w}) \\ & \quad \times \mathbf{R}_k^{-1} (\mathbf{y}_k - g_k(\bar{\mathbf{x}}_{0,b}) - \mathbf{\Gamma}_k \mathbf{w}), \end{aligned} \quad (18)$$

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

The approximate objective function  $\hat{J}_w$  is a quadratic function of  $\mathbf{w}$ , and it no longer contains the Jacobian of the function  $g_k$ . 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  $\mathbf{w}$  becomes the following:

$$\nabla_w \hat{J}_w = \mathbf{w} - \sum_{k=1}^K \left( \mathbf{\Gamma}_k^T \mathbf{R}_k^{-1} [\mathbf{y}_k - g_k(\bar{\mathbf{x}}_{0,b}) - \mathbf{\Gamma}_k \mathbf{w}] \right) \quad (19)$$

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

$$\mathbf{H}_{\hat{J}_w} = \mathbf{I} + \sum_k \left[ \mathbf{\Gamma}_k^T \mathbf{R}_k^{-1} \mathbf{\Gamma}_k \right]. \quad (20)$$

We can thus immediately find the value of  $\mathbf{w}$  when maximizing  $\hat{J}_w$  as follows:

$$\begin{aligned} \hat{\mathbf{w}} & = \left( \mathbf{I} + \sum_k \mathbf{\Gamma}_k^T \mathbf{R}_k^{-1} \mathbf{\Gamma}_k \right)^{-1} \\ & \quad \times \sum_k \left( \mathbf{\Gamma}_k^T \mathbf{R}_k^{-1} [\mathbf{y}_k - g_k(\bar{\mathbf{x}}_{0,b})] \right). \end{aligned} \quad (21)$$

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

$$\hat{\mathbf{x}}_0 = \bar{\mathbf{x}}_{0,b} + \mathbf{X}_{0,b} \hat{\mathbf{w}}. \quad (22)$$

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  $\mathbf{g}_k$ , 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 the matrix  $\mathbf{G}_k \mathbf{X}_{0,b}$  by using the ensemble. This approximation is based on the first-order approximation shown in Eq. (14). Where  $\mathbf{x}_0$  exhibits high uncertainty and  $\|\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}\|$  can be large, this approximation appears to be invalid. Therefore, it is not guaranteed that the estimate with Eq. (22) provides the optimal  $\mathbf{x}_0$  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 that 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:

$$J_\ell(\mathbf{x}_0) = -\frac{1}{2} \sum_{k=1}^K [\mathbf{y}_k - \mathbf{g}_k(\mathbf{x}_0)]^T \mathbf{R}^{-1} [\mathbf{y}_k - \mathbf{g}_k(\mathbf{x}_0)], \quad (23)$$

instead of the log-posterior density function in Eq. (10). Maximization of the Bayesian-type objective function in Eq. (10) will be discussed in Sect. 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,  $\mathbf{y} = \mathbf{y}_{1:K}$  and  $\mathbf{g}(\mathbf{x}_0) = \mathbf{g}_{0:K}(\mathbf{x}_0)$ . The covariance matrices  $\mathbf{R}_1, \dots, \mathbf{R}_K$  are also combined into one block diagonal matrix  $\mathbf{R}$ , which satisfies the following:

$$\mathbf{y}^T \mathbf{R}^{-1} \mathbf{y} = \sum_{k=1}^K \mathbf{y}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k. \quad (24)$$

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

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

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

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

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

$$\mathbf{X}_{m-1} = \frac{1}{\sqrt{N}} \times \begin{pmatrix} \mathbf{x}_{0,m-1}^{(1)} - \bar{\mathbf{x}}_{0,m-1} & \dots & \mathbf{x}_{0,m-1}^{(N)} - \bar{\mathbf{x}}_{0,m-1} \end{pmatrix}, \quad (27)$$

$$\mathbf{\Gamma}_{m-1} = \frac{1}{\sqrt{N}} \times \begin{pmatrix} \mathbf{g}(\mathbf{x}_{0,m-1}^{(1)}) - \mathbf{g}(\bar{\mathbf{x}}_{0,m-1}) & \dots & \mathbf{g}(\mathbf{x}_{0,m-1}^{(N)}) - \mathbf{g}(\bar{\mathbf{x}}_{0,m-1}) \end{pmatrix}, \quad (28)$$

we consider the following  $m$ th objective function:

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

where  $\sigma_m$  is an appropriately chosen parameter. This objective function  $\check{J}_{\ell,m}$  is maximized when, in the following:

$$\begin{aligned} \hat{\mathbf{w}}_m &= \left( \sigma_m^2 \mathbf{I} + \mathbf{\Gamma}_{m-1}^T \mathbf{R}^{-1} \mathbf{\Gamma}_{m-1} \right)^{-1} \\ &\times \left( \mathbf{\Gamma}_{m-1}^T \mathbf{R}^{-1} [\mathbf{y} - \mathbf{g}(\bar{\mathbf{x}}_{0,m-1})] \right), \end{aligned} \quad (30)$$

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

$$\bar{\mathbf{x}}_{0,m} = \bar{\mathbf{x}}_{0,m-1} + \mathbf{X}_{m-1} \bar{\mathbf{w}}_m. \quad (31)$$

Unless converged, members of the next ensemble are generated in the neighbor of  $\bar{\mathbf{x}}_{0,m}$  so that  $\|\mathbf{x}_{0,m}^{(i)} - \bar{\mathbf{x}}_{0,m}\|^2$  is small for each  $i$ , and we proceed to the next iteration. By iterating the above procedures until convergence, the optimal  $\hat{\mathbf{x}}_0$  which maximizes  $J_\ell$  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 obtaining a matrix  $\mathbf{X}_m$  as a scalar multiple of  $\mathbf{X}_{0,b}$ , as follows:

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

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

$$\mathbf{X}_m = \frac{1}{\sqrt{N}} \begin{pmatrix} \mathbf{x}_{0,m}^{(1)} - \bar{\mathbf{x}}_{0,m} & \cdots & \mathbf{x}_{0,m}^{(N)} - \bar{\mathbf{x}}_{0,m} \end{pmatrix}. \quad (33)$$

As discussed later,  $\alpha_m$  should be taken to be so small that a linear approximation is valid over the range of the ensemble dispersion. The value of  $\alpha_m$  can be fixed at a small value. Otherwise,  $\alpha_m$  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  $\mathbf{X}_m$  as outlined by Bocquet and Sakov (2012); that is, in the following:

$$\mathbf{X}_m = \mathbf{X}_{m-1} \mathbf{T}_m, \quad (34)$$

where  $\mathbf{T}_m$  is the ensemble transform matrix given as follows:

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

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

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

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

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 Sect. 6.

---

**Algorithm 1** Iterative algorithm for maximizing the log-likelihood function  $J_\ell$ .

---

Give an initial estimate of  $\mathbf{x}_0$ ,  $\bar{\mathbf{x}}_{0,0}$ .

Give an initial square root of the covariance,  $\mathbf{X}_0$ .

Let  $m = 1$ .

**while** unconverged **do**

Generate an ensemble  $\{\mathbf{x}_{0,m-1}^{(i)}\}_{i=1}^N$  in the neighbor of  $\bar{\mathbf{x}}_{0,m-1}$  by using either of Eq. (32), (34), or (36).

Obtain  $\mathbf{X}_{m-1}$  and  $\mathbf{\Gamma}_{k,m-1}$  in Eqs. (27) and (28).

Compute  $\hat{\mathbf{w}}_m$  in Eq. (30).

Compute the  $m$ -th mean vector  $\bar{\mathbf{x}}_{0,m}$  according to Eq. (31).

Compute the matrix  $\mathbf{X}_m$  according to Eq. (34).

Let  $m := m + 1$

**end while**

---

## 5 Rationale of the algorithm

Equation (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  $\{\mathbf{x}_{0,0}^{(1)}, \dots, \mathbf{x}_{0,0}^{(N)}\}$ . In particular, if  $\sigma_m^2$  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  $\mathbf{g}_k$ ,  $\mathbf{G}_{m-1}$ . Since the above iterative algorithm does not directly use  $\mathbf{G}_{m-1}$ , it would not be trivial to determine how the convergence of this algorithm is achieved. This issue is explored in this section.

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

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

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

we assume the following:

$$\mathbf{x}_0 = \bar{\mathbf{x}}_{0,m-1} + \mathbf{X}_{m-1} \mathbf{w}_m, \quad (38)$$

where  $\mathbf{X}_{m-1}$  is obtained by Eq. (27) given the ensemble  $\{\mathbf{x}_{0,m-1}^{(1)}, \dots, \mathbf{x}_{0,m-1}^{(N)}\}$ . We then have the following:

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

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

$$\mathbf{G}_{m-1} \mathbf{X}_{m-1} \approx \mathbf{\Gamma}_{m-1}, \quad (40)$$

where, in the following:

$$\begin{aligned} \mathbf{\Gamma}_{m-1} &= \frac{1}{\sqrt{N}} \\ &\times (\mathbf{g}(\mathbf{x}_{0,m-1}^{(1)}) - \mathbf{g}(\bar{\mathbf{x}}_{0,m-1}) \quad \dots \quad \mathbf{g}(\mathbf{x}_{0,m-1}^{(N)}) - \mathbf{g}(\bar{\mathbf{x}}_{0,m-1})). \end{aligned} \quad (41)$$

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

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

If we consider a vector  $\mathbf{\Gamma}_{m-1} \mathbf{w}_m$ , it becomes the following:

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

If  $\mathbf{G}_{m-1} \mathbf{X}_{m-1} \mathbf{w}_m$ , which is contained in the first-order term in Eq. (39), is approximated by  $\mathbf{\Gamma}_{m-1} \mathbf{w}_m$ , then 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  $\|\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_{0,m-1}\|$  is very small, any  $\mathbf{x}_0$ , which may have a large uncertainty, can be represented by taking the scale of  $\|\mathbf{w}_m\|$  to be large according to Eq. (38). The nonlinear terms of the right-hand side of Eq. (43) are of the order of  $\|\mathbf{w}_m\|$ , while they are of the order of  $\|\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_{0,m-1}\|^2$  or higher 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 the following:

$$\mathbf{\Gamma}_{m-1} \mathbf{w}_m \approx \mathbf{G}_{m-1} \mathbf{X}_{m-1} \mathbf{w}_m. \quad (44)$$

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

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

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

$$J_\ell(\bar{\mathbf{x}}_{0,m-1} + \mathbf{X}_{m-1} \mathbf{w}_m) \approx J_{\ell, \mathbf{w}_m}(\mathbf{w}_m). \quad (46)$$

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  $\|\mathbf{w}_m\|^2$  and of the order of  $\|\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_{0,m-1}\|^2$ . 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  $\phi(\mathbf{z})$  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 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  $\mathbf{z}_{m-1}$ ,  $\psi(\mathbf{z}_0|\mathbf{z}_{m-1})$ , is chosen to satisfy the following conditions:

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

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

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

the following:

$$\phi(z_{m-1}) = \psi(z_{m-1}|z_{m-1}) \leq \psi(z_m|z_{m-1}) \leq \phi(z_m), \quad (48)$$

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

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

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

$$\begin{aligned} J_{\ell, w_m}^\dagger(w_m|\bar{x}_{0, m-1}) &= -\frac{\sigma_m^2}{2} w_m^T w_m \\ &\quad - \frac{1}{2} [y - g(\bar{x}_{0, m-1})]^T R^{-1} [y - g(\bar{x}_{0, m-1})] \\ &\quad + [y - g(\bar{x}_{0, m-1})]^T R^{-1} \Gamma_{m-1} w_m \\ &\quad - \frac{1}{2} w_m^T \Gamma_{m-1}^T R^{-1} \Gamma_{m-1} w_m \\ &= -\frac{\sigma_m^2}{2} w_m^T w_m - \frac{1}{2} [y - g(\bar{x}_{0, m-1}) - \Gamma_{m-1} w_m]^T \\ &\quad \times R^{-1} [y - g(\bar{x}_{0, m-1}) - \Gamma_{m-1} w_m], \end{aligned} \quad (49)$$

which is a similar treatment to Böhning and Lindsay (1988). For a given  $\Delta$ , we can take  $\sigma_m^2$  so that the following inequality holds over  $\|w_m\| < \Delta$ :

$$\begin{aligned} -\frac{\sigma_m^2}{2} w_m^T w_m &\leq \frac{1}{4} w_m^T X_{m-1}^T \\ &\quad \times \left[ (y - g(\bar{x}_{0, m-1}))^T R^{-1} (\nabla^2 g) \right] \\ &\quad \times X_{m-1} w_m + O(\|w_m\|^3), \end{aligned} \quad (50)$$

where the equality holds if  $w_m = \mathbf{0}$ . If  $\sigma_m^2$  is chosen so that the inequality (50) is satisfied,  $J_{\ell, w}^\dagger$  satisfies the following:

$$\begin{aligned} J_{\ell, w_m}^\dagger(w_m|\bar{x}_{0, m-1}) &\leq J_{\ell, w_m}(w_m) \\ &\approx J_{\ell}(\bar{x}_{0, m-1} + X_{m-1} w_m), \end{aligned} \quad (51)$$

$$J_{\ell, w_m}^\dagger(\mathbf{0}|\bar{x}_{0, m-1}) = J_{\ell, w_m}(\mathbf{0}) = J_{\ell}(\bar{x}_{0, m-1}). \quad (52)$$

This means that  $J_{\ell, w_m}^\dagger$  can be used as a surrogate function for maximizing  $J_{\ell, w_m}(w_m)$  according to the MM algorithm. Since Eq. (49) is the same as Eq. (29), the maximum of  $J_{\ell, w_m}^\dagger$

is achieved when  $w_m = \hat{w}_m$  where  $\hat{w}_m$  is given by Eq. (30). Obviously,  $\hat{w}_m$  satisfies the following inequality:

$$J_{\ell, w_m}^\dagger(\mathbf{0}|\bar{x}_{0, m-1}) \leq J_{\ell, w_m}^\dagger(\hat{w}_m|\bar{x}_{0, m-1}), \quad (53)$$

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

$$\begin{aligned} J_{\ell}(\bar{x}_{0, m-1}) &= J_{\ell, w_m}(\mathbf{0}) \leq J_{\ell, w_m}(\hat{w}_m) \\ &\approx J_{\ell}(\bar{x}_{0, m-1} + X_{m-1} \hat{w}_m) = J_{\ell}(\bar{x}_{0, m}), \end{aligned} \quad (54)$$

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

The above discussion is valid regardless of the choice of the ensemble  $\{x_{0, m}^{(1)}, \dots, x_{0, m}^{(N)}\}$  in each iteration as far as the approximation in Eq. (40) is applicable. This suggests that 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), then the ensemble is confined within a particular subspace spanned by the initial ensemble, and  $\bar{x}_{0, m-1}$  would converge to a stationary point in this subspace. According to Eq. (37), if the nonlinearity of  $g$  is not severe when  $(\nabla^2 g)$  is not dominant, the Hessian of  $J_{\ell}$  is negative definite in a region where  $\|y - g(\bar{x}_{0, m-1})\|$  is small enough. This suggests that the iterative algorithm in Sect. 4 would attain at least a local maximum of  $J_{\ell}$  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,  $J_{\ell}$  would increase until a point which can be regarded as a stationary point in any  $N$ -dimensional subspace, and  $\bar{x}_{0, m-1}$  would thus converge to a local maximum in the full vector space after infinite iterations.

Based on the foregoing, convergence to a local maximum of the objective function  $J_{\ell}$  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  $J_{\ell}$  is a 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  $J_{\ell}$ , it might be effectual to start with an ensemble with a large spread to approach the global maximum. An ensemble with a 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  $\bar{\mathbf{x}}_{0,m-1}$  for obtaining the  $m$ th estimate in Eq. (31). On the other hand, in many studies, the ensemble mean of simulation runs  $\{\mathbf{g}(\mathbf{x}_{0,m-1}^{(i)})\}$  is used as a

substitute for  $\mathbf{g}(\bar{\mathbf{x}}_{0,m-1})$ . If the ensemble mean  $\mathbf{g}(\mathbf{x}_{0,m-1}^{(i)})$  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  $\bar{\mathbf{x}}_{0,m-1}$ . 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  $\mathbf{g}(\bar{\mathbf{x}}_{0,m-1})$ . 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  $\sigma_m^2$ . A sufficiently large  $\sigma_m^2$  guarantees that the  $m$ th estimate  $\bar{\mathbf{x}}_{0,m}$  is better than the previous estimate  $\bar{\mathbf{x}}_{0,m-1}$ , and hence, convergence is stable. However, convergence speed will be degraded with large  $\sigma_m^2$  because  $\bar{\mathbf{x}}_{0,m}$  is strongly constrained by the penalty weighted with  $\sigma_m^2$ . 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 then, in the following:

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

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

## 6 Bayesian form

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

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

which is the same as Eq. (10), although the vectors of the whole time sequence are combined into a single vector for

each  $\mathbf{y}$  and  $\mathbf{g}(\mathbf{x}_0)$  as in Eq. (23). Equation (56) is proportional to the log-posterior distribution when  $p(\mathbf{x}_0)$  and  $p(\mathbf{y}|\mathbf{x}_0)$  are assumed to be Gaussian. The Taylor expansion of Eq. (56) is as follows:

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

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

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

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

$$\begin{aligned}
J_{w_m}^\dagger(w_m|\bar{x}_{0,m-1}) &= -\frac{\sigma_m^2}{2} w_m^T w_m \\
&\quad - \frac{1}{2} (\bar{x}_{0,m-1} - \bar{x}_{0,b} + \mathbf{X}_{m-1} w_m)^T \\
&\quad \times \mathbf{P}_{0,b}^{-1} (\bar{x}_{0,m-1} - \bar{x}_{0,b} + \mathbf{X}_{m-1} w_m) \\
&\quad - \frac{1}{2} [y - g(\bar{x}_{0,m-1})]^T \mathbf{R}^{-1} [y - g(\bar{x}_{0,m-1})] \\
&\quad + [y - g(\bar{x}_{0,m-1})]^T \mathbf{R}^{-1} \Gamma_{m-1} w_m \\
&\quad - \frac{1}{2} w_m^T \Gamma_{m-1}^T \mathbf{R}^{-1} \Gamma_{m-1} w_m \\
&= -\frac{\sigma_m^2}{2} w_m^T w_m - \frac{1}{2} (\bar{x}_{0,m-1} - \bar{x}_{0,b})^T \\
&\quad \times \mathbf{P}_{0,b}^{-1} (\bar{x}_{0,m-1} - \bar{x}_{0,b}) \\
&\quad - \frac{1}{2} [y - g(\bar{x}_{0,m-1})]^T \mathbf{R}^{-1} [y - g(\bar{x}_{0,m-1})] \\
&\quad - (\bar{x}_{0,m-1} - \bar{x}_{0,b})^T \mathbf{P}_{0,b}^{-1} \mathbf{X}_{m-1} w_m \\
&\quad + [y - g(\bar{x}_{0,m-1})]^T \mathbf{R}^{-1} \Gamma_{m-1} w_m \\
&\quad - \frac{1}{2} w_m^T \mathbf{X}_{m-1}^T \mathbf{P}_{0,b}^{-1} \mathbf{X}_{m-1} w_m \\
&\quad - \frac{1}{2} w_m^T \Gamma_{m-1}^T \mathbf{R}^{-1} \Gamma_{m-1} w_m, \tag{59}
\end{aligned}$$

which satisfies the following conditions:

$$J_{w_m}^\dagger(w_m|\bar{x}_{0,m-1}) \leq J_{w_m}(w_m) \approx J(\bar{x}_{0,m-1} + \mathbf{X}_{m-1} w_m), \tag{60}$$

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

This function is maximized when, in the following:

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

The  $m$ th estimate for  $\mathbf{x}_0$  is obtained as follows:

$$\bar{x}_{0,m} = \bar{x}_{0,m-1} + \mathbf{X}_{m-1} \hat{w}_m. \tag{63}$$

Similar to Eq. (54), we obtain the following:

$$\begin{aligned}
J(\bar{x}_{0,m-1}) &= J_{w_m}(\mathbf{0}) \leq J_{w_m}(\hat{w}_m) \\
&\approx J(\bar{x}_{0,m-1} + \mathbf{X}_{m-1} \hat{w}_m) = J_\ell(\bar{x}_{0,m}). \tag{64}
\end{aligned}$$

Thus,  $\bar{x}_{0,m}$  is a better estimate than  $\bar{x}_{0,m-1}$  if the approximation in Eq. (44) is valid. Generating the  $(m+1)$ th ensemble around  $\bar{x}_{0,m}$ , 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 Sect. 4. Eq. (32) or (34)

is convenient for practical problems because we can avoid computing the inverse of  $\mathbf{P}_{0,b}$  in Eq. (62). When Eq. (32) is used for updating the ensemble, we can easily avoid computing the inverse of  $\mathbf{P}_{0,b}$  by drawing initial ensemble members from the prior distribution  $\mathcal{N}(\mathbf{x}_0; \mathbf{x}_{0,b}, \mathbf{P}_{0,b})$ . If initial ensemble members  $\{\mathbf{x}_{0,0}^{(1)}, \dots, \mathbf{x}_{0,0}^{(N)}\}$  obey the prior distribution, we can use the same approximation as Eq. (13); that is, in the following:

$$\mathbf{X}_{0,b}^T \mathbf{P}_{0,b}^{-1} \mathbf{X}_{0,b} \approx \mathbf{I}. \tag{65}$$

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

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

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

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

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

As described in the previous section, the use of Eq. (32) or (34) confines the estimate  $\bar{x}_{0,m-1}$  within the subspace spanned by the initial ensemble. On the other hand, Eq. (36) enables us to seek the optimal value of  $\mathbf{x}_0$  in a different subspace in each iteration. We can then obtain the local maximum in the full vector space if  $\mathbf{Q}_m$  is taken to be full rank. It appears that a similar approximation to Eq. (67) is applicable if  $\mathbf{Q}_m$  is taken to be a scalar matrix of  $\mathbf{P}_{0,b}$ . 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 inverse of  $\mathbf{P}_{0,b}$  in general. Nonetheless, if  $\mathbf{P}_{0,b}^{-1}$  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:

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

for  $m = 1, \dots, M$ , 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  $\zeta = 5$ .

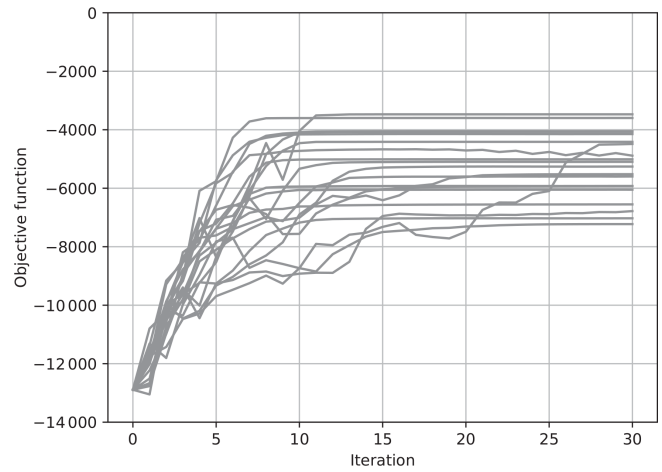
We compare two ensemble updating methods of Eqs. (32) and (36). In applying Eq. (32), the initial ensemble  $\{\mathbf{x}_{0,0}^{(1)}, \dots, \mathbf{x}_{0,0}^{(N)}\}$  was drawn from a Gaussian distribution  $\mathcal{N}(\mathbf{0}, \varepsilon^2 \mathbf{I})$ , where  $\varepsilon = 5 \times 10^{-6}$  and  $\mathbf{X}_0$  was obtained as follows:

$$\mathbf{X}_0 = \frac{1}{\sqrt{N}} \begin{pmatrix} \mathbf{x}_{0,0}^{(1)} - \bar{\mathbf{x}}_{0,0} & \dots & \mathbf{x}_{0,0}^{(N)} - \bar{\mathbf{x}}_{0,0} \end{pmatrix}. \quad (69)$$

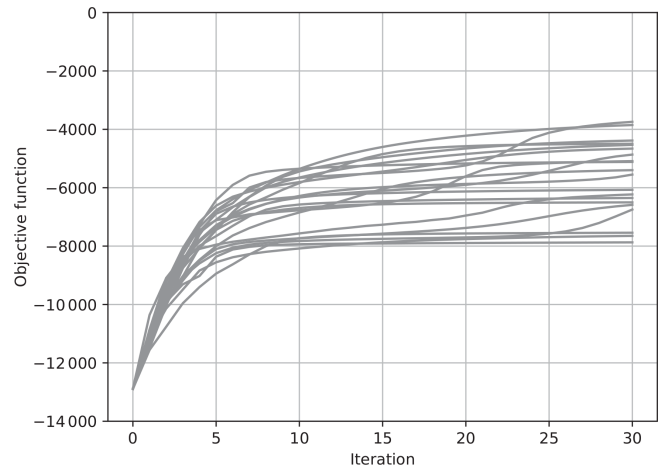
The matrix  $\mathbf{X}_m$  for each iteration was fixed at  $\mathbf{X}_m = \mathbf{X}_0$ , which corresponds to the setting in Eq. (32) with  $\alpha_m = 1$ . The discussion in Sect. 5 suggests that the penalty parameter  $\sigma_m^2$  should be determined according to Eq. (55). Although  $(\nabla^2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

$$\sigma_m^2 = \delta^2 \sqrt{(\mathbf{y}_K - \mathbf{g}_K(\hat{\mathbf{x}}_{0,m-1}))^T \mathbf{R}_K^{-1} (\mathbf{y}_K - \mathbf{g}_K(\hat{\mathbf{x}}_{0,m-1}))} \\ \times \text{tr}(\mathbf{\Gamma}_{m-1}^T \mathbf{R}^{-1} \mathbf{\Gamma}_{m-1}), \quad (70)$$

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  $\mathbf{y}_K - \mathbf{g}_K(\hat{\mathbf{x}}_{0,m-1})$  was multiplied in order that  $\sigma_m^2$  was roughly proportional to  $\|\mathbf{y} - \mathbf{g}(\bar{\mathbf{x}}_{0,m-1})\|$ , and  $\text{tr}(\mathbf{\Gamma}_{m-1}^T \mathbf{R}^{-1} \mathbf{\Gamma}_{m-1})$  was for representing the variance of the ensemble.

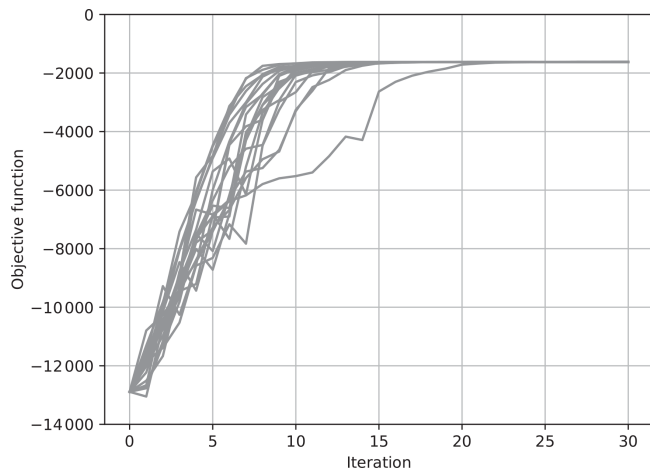


**Figure 1.** The value of the objective function  $J$  for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (32) with  $\delta = 1.5 \times 10^{-3}$ .



**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  $\delta = 1.5 \times 10^{-2}$ .

Figures 1 and 2 show 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 Sect. 5, monotonic convergence is achieved when  $\sigma_m^2$  is taken to be large enough. However, convergence speed becomes slow when  $\sigma_m^2$  is large. The results in Figs. 1 and 2 thus confirmed our discussion on the convergence. However, the results shown

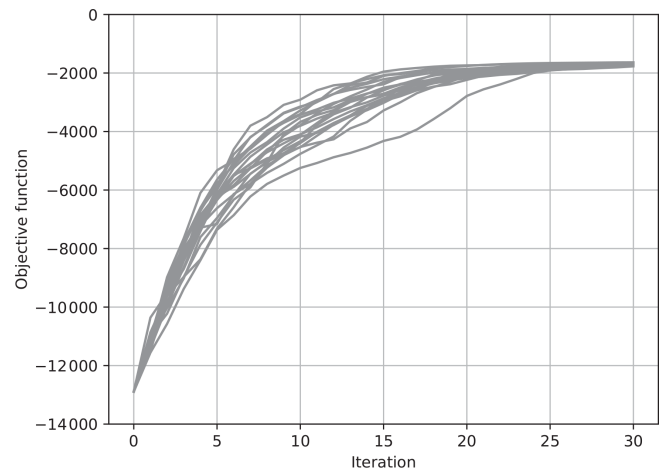


**Figure 3.** The value of the objective function  $J$  for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (36) with  $\delta = 1.5 \times 10^{-3}$ .

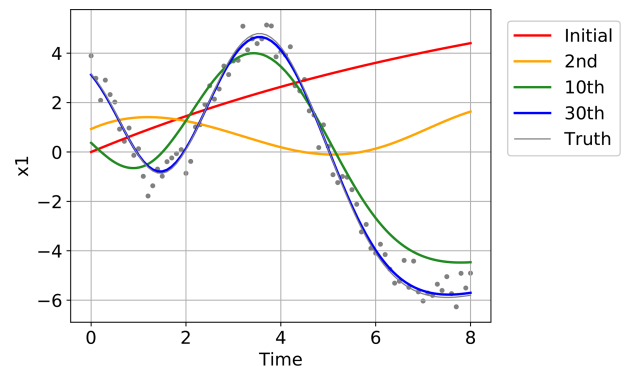
in Figs. 1 and 2 did not converge to the same value, which means the results depended on the seeds of pseudo random numbers. This would indicate that a local maximum within a subspace spanned by the ensemble does not match the maximum in the full state vector space, and that the value of the local maximum depends on the subspace.

Figures 3 and 4 show 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 in  $J$  tended to be sharp, while it was not monotonic. On the other hand, when  $\delta = 1.5 \times 10^{-2}$ , the increase in  $J$  was gradual but monotonic. In contrast with the results in Figs. 3 and 4, the values of  $J$  in different trials converged to the same value after about 15 iterations, as in the case with  $\delta = 1.5 \times 10^{-3}$  shown in Fig. 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 case with  $\delta = 1.5 \times 10^{-2}$  after about 80 iterations in all of the 20 trials (not shown). These results show 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.

Figure 5 shows the convergence of the estimated time series to the true time series for one of the 40 state variables,  $x_1$ , in one of the 20 trials in Fig. 4. The red line indicates the initial guess obtained by running the Lorenz 96 model started at  $\bar{x}_{0,0}$ . The yellow, green, and blue lines show the estimates after the second, 10th, and 30th iterations, respectively. The truth is indicated with the gray line, and the time series of the synthetic observations used in this experiment are overplotted with gray dots. Although the initial trajectory (red line) showed was obviously dissimilar to the true trajectory (gray line), the estimate was improved by repeating the iterations as also shown in Fig. 4. After 30 iterations, the es-



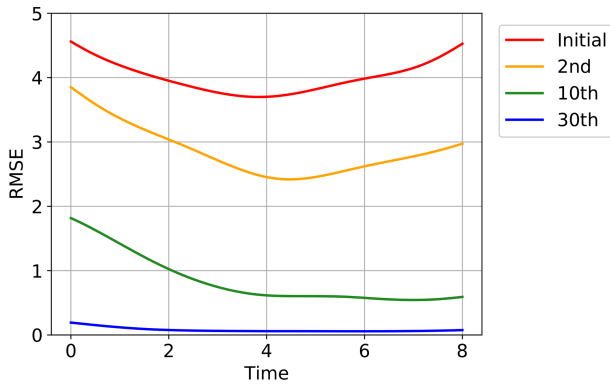
**Figure 4.** The value of the objective function  $J$  for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (36) with  $\delta = 1.5 \times 10^{-2}$ .



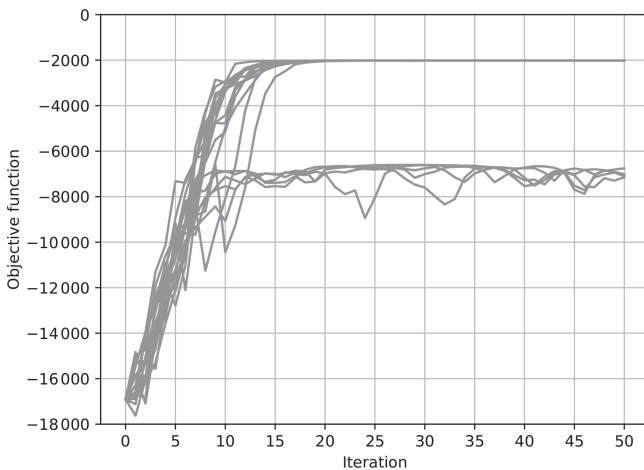
**Figure 5.** The temporal evolution started at the initial guess (red line), and the estimated evolutions after the second (yellow line), 10th (green line), and 30th iterations (blue line) in one of the 20 trials in Fig. 4. The truth is indicated with the gray line, and the time series of the synthetic observations used in this experiment are overplotted with gray dots.

timate was very close to the truth, and the temporal evolution was well reproduced. Figure 6 shows the root mean square errors of the estimates, which means the root mean squares of the differences between the estimates and the true values over all the 40 variables at each time step in the same trial as in Fig. 5. Again, the red line indicates the initial guess, and the yellow, green, and blue lines show the estimates after the second, 10th, and 30th iterations, respectively. The errors were certainly reduced over the period of the assimilation by the iterations.

In order to closely investigate the effect of  $\sigma_m^2$ , we conducted additional experiments for a case in which nonlinearity is a little stronger. While Figs. 3 and 4 show the results when the assimilation window was taken as  $0 < t \leq 8$ , Figs. 7 and 8 show the results with a little longer assimilation window,  $0 < t \leq 10$ . Although the other settings were the same



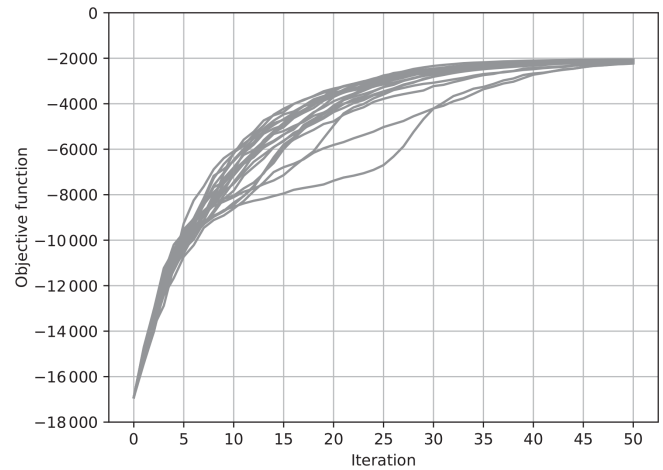
**Figure 6.** The root mean square errors over all the 40 variables at each time step for the initial guess (red line), the second iteration (yellow line), the 10th iteration (green line), and the 30th iteration (blue line) in the same trial as Fig. 5.



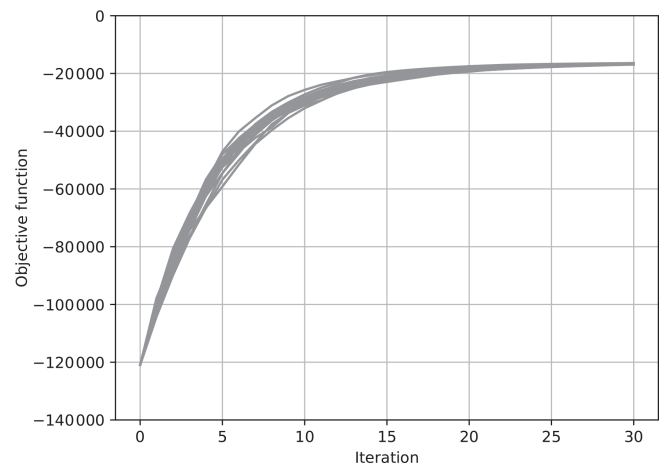
**Figure 7.** The value of the objective function  $J$  for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (36) with  $\delta = 1.5 \times 10^{-3}$ , and the ensemble window was taken as  $0 < t \leq 10$ .

as Figs. 3 and 4, the effect of the nonlinearity on the objective function  $J$  was a little more severe 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  $\delta$  was as large as  $1.5 \times 10^{-2}$ , the  $J$  value converged to the same value after about 50 iterations in all of the 20 trials. As discussed in Sect. 5, a sufficiently large  $\sigma_m^2$  guarantees that the estimate is improved in each iteration. Although the convergence speed becomes worse, a 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 10 times higher. Figure 9 shows the result with 400 variables. The assimilation

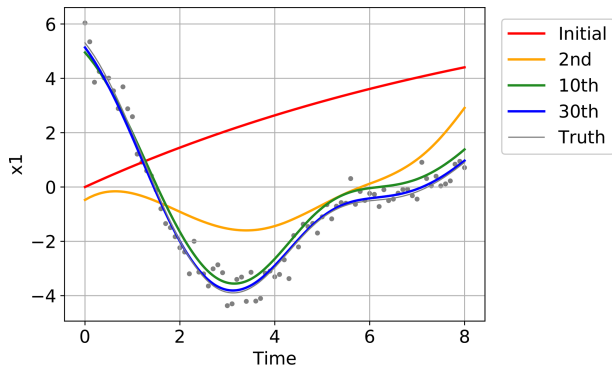


**Figure 8.** The value of the objective function  $J$  for each iteration for 20 trials of the estimation. The ensemble was updated using Eq. (36) with  $\delta = 1.5 \times 10^{-2}$ , and the ensemble window was taken as  $0 < t \leq 10$ .

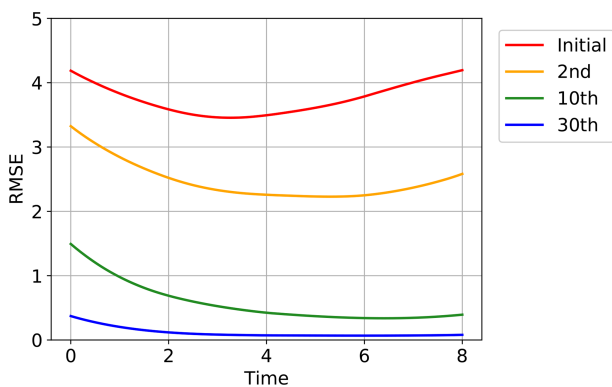


**Figure 9.** The value of the objective function  $J$  for each iteration for 20 trials of the estimation for the 400 dimensional system. The ensemble was updated using Eq. (36) with  $\delta = 1.5 \times 10^{-3}$ .

was taken as  $0 < t \leq 8$ , and  $\delta$  was set at  $1.5 \times 10^{-3}$ , which is the same as the result in Fig. 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  $\delta$  was taken to be as small as in Fig. 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 for the stable convergence for the high-dimensional Lorenz 96 systems at present.



**Figure 10.** The temporal evolution started at the initial guess (red line), and the estimated evolutions at the second (yellow line), 10th (green line), and 30th iterations (blue line) in one of the 20 trials in Fig. 9. The truth is indicated with the gray line, and the time series of the synthetic observations used in this experiment are overplotted with gray dots.



**Figure 11.** The root mean square errors over all the 400 variables at each time step for the initial guess (red line), the second iteration (yellow line), the 10th iteration (green line), and the 30th iteration (blue line) in the same trial as Fig. 10.

In Fig. 10, we confirmed the convergence of the estimated time series to the true trajectory for one of the 400 state variables,  $x_1$ , in one of the 20 trials in Fig. 9. The red line indicates the initial guess started at  $\bar{x}_{0,0}$ , and the yellow, green, and blue lines show the estimates after the second, 10th, and 30th iterations, respectively. The gray line shows the truth, and the gray dots show the synthetic observations used in this experiment. Similar to Fig. 5, the estimate approached to the truth by repeating the iterations even in this high-dimensional case, and the true evolution was well reproduced after 30 iterations. In Fig. 11, the root mean square errors over all the 400 variables at each time step are in the same trial as Fig. 10. Each color corresponds to the respective iteration shown in Fig. 10. It is confirmed that the errors decreased over the period of the assimilation after repeating the iterations.

## 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 where 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  $g$ . 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  $\sigma_m^2$  must be set to be large enough. A sufficiently large  $\sigma_m^2$  would ensure monotonic convergence, although convergence speed would become poorer with too large a  $\sigma_m^2$ . The effect of this penalty term has also been experimentally confirmed in Sect. 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  $\sigma_m^2$  would be an important issue. Although it was determined according to Eq. (70) in Sect. 7, Eq. (70) still requires tuning of the parameter  $\delta$ . However, it is not necessary to finely tune  $\delta$  because  $\delta$  would not have a crucial effect on the performance of the algorithm. It would thus be enough to roughly determine  $\delta$ . In addition, one could check whether the objective function  $J$  increases or not at each iteration just by running one forward simulation initialized at  $\bar{x}_{0,m}$ . In the iterative algorithm, the most computational cost is spent on 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 of tuning  $\delta$  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  $x_0$ . 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  $\sigma_m^2$  is sufficiently large, then 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, as indicated in Eq. (36), the optimal solution would be sought in a different subspace in each iteration, and the estimate would converge to a local maximum in the full vector space after infinite iterations. It should be noted that this paper has not assessed how well the method with a random ensemble generation with Eq. (36) works in practical high-dimensional problems, although it is, in theory, applicable as far as the inverse of  $P_{0,b}$  is available.

Further research would be required to clarify its performance in high-dimensional geophysical models in order to reinforce this study.

Compared with the adjoint method, which is a conventional variational method for 4D 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.

## 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  $\mathbf{P}_{0,b}$  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) as follows:

$$\mathbf{X}_m = \mathbf{X}_{m-1} \mathbf{T}_m, \quad (\text{A1})$$

the transform matrix  $\mathbf{T}_m$  should be given as follows:

$$\mathbf{T}_m = \mathbf{U}_m (\mathbf{I} + \mathbf{\Lambda}_m)^{-\frac{1}{2}} \mathbf{U}_m^T, \quad (\text{A2})$$

where  $\mathbf{U}_m$  and  $\mathbf{\Lambda}_m$  are obtained by the following eigenvalue decomposition:

$$\mathbf{U}_m \mathbf{\Lambda}_m \mathbf{U}_m^T = \sigma_m^{-2} \left( \mathbf{X}_{m-1}^T \mathbf{P}_{0,b}^{-1} \mathbf{X}_{m-1} + \mathbf{\Gamma}_{m-1}^T \mathbf{R}^{-1} \mathbf{\Gamma}_{m-1} \right). \quad (\text{A3})$$

If  $\mathbf{X}_{m-1}$  is obtained according to Eq. (A1), as follows:

$$\mathbf{X}_{m-1} = \mathbf{X}_{m-2} \mathbf{T}_{m-1} = \mathbf{X}_0 \mathbf{T}_1 \mathbf{T}_2 \cdots \mathbf{T}_{m-1}. \quad (\text{A4})$$

Defining the matrix  $\mathbf{C}_{m-1}$  as follows:

$$\mathbf{C}_{m-1} = \mathbf{T}_1 \mathbf{T}_2 \cdots \mathbf{T}_{m-1}, \quad (\text{A5})$$

$\mathbf{X}_{m-1}$  can be written as follows:

$$\mathbf{X}_{m-1} = \mathbf{X}_0 \mathbf{C}_{m-1}. \quad (\text{A6})$$

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

$$\mathbf{X}_{m-1}^T \mathbf{P}_{0,b}^{-1} \mathbf{X}_{m-1} = \mathbf{C}_{m-1}^T \mathbf{C}_{m-1}. \quad (\text{A7})$$

The  $m$ th estimate is broken down as follows:

$$\begin{aligned} \bar{\mathbf{x}}_{0,m-1} &= \bar{\mathbf{x}}_{0,m-2} + \mathbf{X}_{m-2} \hat{\mathbf{w}}_{m-1} = \bar{\mathbf{x}}_{0,b} + \sum_{i=1}^{m-1} \mathbf{X}_{i-1} \hat{\mathbf{w}}_i \\ &= \bar{\mathbf{x}}_{0,b} + \mathbf{X}_0 \sum_{i=1}^{m-1} \mathbf{C}_{i-1} \hat{\mathbf{w}}_i, \end{aligned} \quad (\text{A8})$$

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

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

then Eq. (A8) becomes the following:

$$\bar{\mathbf{x}}_{0,m-1} = \bar{\mathbf{x}}_{0,b} + \mathbf{X}_0 \boldsymbol{\xi}_{m-1}, \quad (\text{A10})$$

and we obtain the following:

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

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

$$\begin{aligned} \hat{\mathbf{w}}_m &= \left( \sigma_m^2 \mathbf{I} + \mathbf{C}_{m-1}^T \mathbf{C}_{m-1} + \mathbf{\Gamma}_{m-1}^T \mathbf{R}^{-1} \mathbf{\Gamma}_{m-1} \right)^{-1} \\ &\quad \times \left( \mathbf{\Gamma}_{m-1}^T \mathbf{R}^{-1} [\mathbf{y} - \mathbf{g}(\bar{\mathbf{x}}_{0,m-1})] - \mathbf{C}_{m-1}^T \boldsymbol{\xi}_{m-1} \right). \end{aligned} \quad (\text{A12})$$

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.

---

Give an initial estimate of  $\mathbf{x}_0, \bar{\mathbf{x}}_{0,0}$ .

Give an initial square root of the covariance,  $\mathbf{X}_0$ .

Let  $\mathbf{C}_0 = \mathbf{I}$  and  $\boldsymbol{\xi}_0 = \mathbf{0}$ .

Let  $m = 1$ .

**while** unconverged **do**

    Generate an ensemble  $\{\mathbf{x}_{0,m}\}_{i=1}^N$  around with a mean of  $\bar{\mathbf{x}}_{0,m-1}$  and a variance of  $\mathbf{X}_{m-1} \mathbf{X}_{m-1}^T$ .

    Obtain  $\mathbf{\Gamma}_{k,m-1}$  in Eq. (28).

    Compute  $\hat{\mathbf{w}}_m$  in Eq. (A12).

    Compute the  $m$ -th estimate  $\bar{\mathbf{x}}_{0,m}$  according to Eq. (63).

    Compute the matrix  $\mathbf{X}_m$  according to Eq. (A1).

    Let  $\mathbf{C}_m = \mathbf{C}_{m-1} \mathbf{T}_{m-1}$

    Let  $\boldsymbol{\xi}_m = \boldsymbol{\xi}_{m-1} + \mathbf{C}_{m-1} \hat{\mathbf{w}}_m$

    Let  $m := m + 1$

**end while**

---

*Code availability.* The code for reproducing the experimental results shown in Sect. 7 is archived on Zenodo (<https://doi.org/10.5281/zenodo.4420875>; Nakano, 2020).

*Competing interests.* The author declares that there is no conflict of interest.

*Financial support.* This research has been supported by the Japan Society for the Promotion of Science (grant no. 17H01704) and by PRC JSPS CNRS, the Bilateral Joint Research Project “Forecasting the geomagnetic secular variation based on data assimilation”.

*Review statement.* This paper was edited by Amit Apte and reviewed by three anonymous referees.

## References

- Bannister, R. N.: A review of operational methods of variational and ensemble-variational data assimilation, *Q. J. Roy. Meteor. 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. Weather Rev.*, 129, 420–436, 2001.
- Bocquet, M. and Sakov, P.: Combining inflation-free and iterative ensemble Kalman filters for strongly nonlinear systems, *Nonlin. Processes 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 an iterative ensemble Kalman smoother, *Nonlin. Processes 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. Roy. Meteor. 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. Roy. Meteor. 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. Weather 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. Roy. Meteor. 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 ensemble Kalman smoother for nonlinear dynamics, *Mon. Weather 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, 11948–11956, <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. Roy. Meteor. 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, Society for Industrial and Applied Mathematics, Philadelphia, USA, 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. Roy. Meteor. 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. Weather 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. Weather 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. Roy. Meteor. 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, 399–414, 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 4D-EnVar data assimilation, *Earth Planet. Space*, 72, 136, <https://doi.org/10.1186/s40623-020-01253-8>, 2020.
- Nakano, S.: Experimental demonstration of the iterative ensemble-based variational method, *Zenodo*, <https://doi.org/10.5281/zenodo.4420875>, 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, A05208, <https://doi.org/10.1029/2006JA011853>, 2008.
- Nocedal, J. and Wright, S. J.: *Numerical optimization*, 2nd ed., Springer, New York, USA, 2006.
- Raanes, P. N., Stordal, A. S., and Evensen, G.: Revising the stochastic iterative ensemble smoother, *Nonlin. Processes 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. Weather Rev.*, 124, 2898–2913, 1996.
- Yokota, S., Kunii, M., Aonashi, K., and Origuchi, S.: Comparison between four-dimensional LETKF and ensemble-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. Roy. Meteor. Soc.*, 134, 1039–1050, 2008.