Nonlinear Least Squares En4DVar to 4DEnVar Methods for Data Assimilation: Formulation, Analysis, and Preliminary Evaluation

XIANGJUN TIAN AND HONGQIN ZHANG

ICCES, Institute of Atmospheric Physics, Chinese Academy of Sciences, and University of Chinese Academy of Sciences, Beijing, China

XIAOBING FENG

Department of Mathematics, University of Tennessee, Knoxville, Tennessee

YUANFU XIE

NOAA/Earth System Research Laboratory, Boulder, Colorado

(Manuscript received 7 March 2017, in final form 29 September 2017)

ABSTRACT

The En4DVar method is designed to combine the flow-dependent statistical covariance information of EnKF into the traditional 4DVar method. However, the En4DVar method is still hampered by its strong dependence on the adjoint model of the underlying forecast model and by its complexity, maintenance requirements, and the high cost of computer implementation and simulation. The primary goal of this paper is to propose an alternative approach to overcome the main difficulty of the En4DVar method caused by the use of adjoint models. The proposed approach, the nonlinear least squares En4DVar (NLS-En4DVar) method, begins with rewriting the standard En4DVar formulation into a nonlinear least squares problem, which is followed by solving the resulting NLS problem by a Gauss–Newton iterative method. To reduce the computational and implementation complexity of the proposed NLS-En4DVar method, a few variants of the new method are proposed; these modifications make the model cheaper and easier to use than the full NLS-En4DVar method at the expense of reduced accuracy. Furthermore, an improved iterative method based on the comprehensive analysis on the above NLS_i-En4DVar family of methods is also proposed. These proposed NLS_i-En4DVar methods provide more flexible choices of the computational capabilities for the broader and more realistic data assimilation problems arising from various applications. The pros and cons of the proposed NLS_i-En4DVar family of methods are further examined in the paper and their relationships and performance are also evaluated by several sets of numerical experiments based on the Lorenz-96 model and the Advanced Research WRF (ARW) Model, respectively.

1. Introduction

Data assimilation for numerical weather prediction (NWP) has experienced explosive growth and development after ensemble Kalman filter (EnKF; Evensen 1994; Houtekamer et al. 2014) and four-dimensional variational data assimilation (4DVar; Rabier et al. 2000) techniques were successively introduced as two major competing methods for initializing NWP. It is their competition (Kalnay et al. 2007; Lorenc 2003) and their interplay (Gustafsson 2007) that have fueled the rapid growth and development in this area of research over the past 20 years. On one hand, successful applications of 4DVar at several major operational NWP centers (e.g., ECMWF, and those of France, the United Kingdom, Japan, and China) demonstrate that its development has reached an advanced and mature stage. A consensus view point in the NWP community is that 4DVar has advantages in producing highly accurate predictions and having the ability to assimilate

Denotes content that is immediately available upon publication as open access.

Corresponding author: Xiangjun Tian, tianxj@mail.iap.ac.cn

DOI: 10.1175/MWR-D-17-0050.1

asynchronous observations simultaneously. On the other hand, 4DVar has been hampered by its strong dependence on the adjoint model of the forecast model and its complexity, maintenance requirements during operations, and the high costs associated with programming and computer simulations. Moreover, the adjoint models are often very difficult to obtain when the model physics involves substantial "on–off" processes (Xu 1996; Zou et al. 1997; Steward et al. 2012), as these limitations call for improvements and for developing more efficient methods such as EnKF or some nonsmooth optimization approaches (e.g., Karmitsa 2016). In contrast, EnKF has become increasingly competitive with 4DVar due to its capability of providing flow-dependent error estimates from the statistics of the ensemble samples. However, the performance of EnKF could be compromised by sampling errors because of limited sample sizes, which has indeed been a difficult issue throughout the development of EnKF.

Recently much effort has been devoted to developing hybrid methods based on 4DVar and EnKF, which could ideally inherit the advantages of both methods but avoid their deficiencies (Gustafsson 2007; Kalnay et al. 2007; Bowler et al. 2017a,b; Lorenc 2017; Lorenc et al. 2017; Bannister 2017; Amezcua et al. 2017; Goodliff et al. 2017). For example, on the EnKF side, Hunt et al. (2004) extended EnKF to four dimensions in order to allow the use of asynchronous observations, which thus introduced the 4DEnKF method. The 4DEnKF approach can assimilate observations at selected time points just like 4DVar does but without iteration. Evensen and Van Leeuwen (2000) proposed the ensemble Kalman smoother to achieve the same purpose. On the 4DVar side, the "hybrid" (e.g., Clayton et al. 2013), the "En4DVar" (e.g., Zhang et al. 2009), and the "4DEnVar" assimilation methods (Qiu et al. 2007; Liu et al. 2008; Tian et al. 2008; Wang et al. 2010; Tian et al. 2011; Tian and Xie 2012; Tian and Feng 2015) were successively proposed to incorporate the statistical flow-dependent strength of EnKF into 4DVar. It was noted by Lorenc (2013) that the hybrid 4DVar uses a combination of climatological and ensemble covariances and the En4DVar utilizes the ensemble background covariance provided by the parallel EnKF implementation. It should be noted that the adjoint model is often used to carry out the minimization procedure in both the hybrid 4DVar and the En4DVar, but the 4DEnVar approach utilizes the simulated observation perturbations (OPs) to approximate the linearized tangent operator and hence avoids using the adjoint models. Indeed, some variations of the 4DEnVar without iteration are similar to 4DEnKF (Tian et al. 2011; Hunt et al. 2004). This phenomenon further explains the necessity of their (4DVar and EnKF) integration. In comparison, the competitiveness of En4DVar is in its more accurate description of the tangent linear model (when the linearity validity of the linear model is satisfied) while the advantage of 4DEnVar is its ease in programming. A natural question is whether we can achieve a balance between 4DVar and EnKF in terms of performance and accuracy. Such a balance, it seems, only can be achieved through a comprehensive analysis of all the factors involved, including assimilation accuracy, programming difficulty, and computational complexity.

The primary objective of this paper is to address the above question. We begin by reviewing the standard En4DVar formulation and then present its reformulation as a nonlinear least squares (NLS) problem. A Gauss–Newton iterative algorithm is then proposed to solve the reformulated NLS problem. Since the exact implementation of the Gauss–Newton algorithm is computationally expensive, we propose various levels of simplifications of the full Gauss–Newton algorithm with decreasing computational complexity. Among these simplified algorithms, we recovered the NLS-En4DVar algorithm proposed by Tian and Feng (2015). If only one iteration is retained in the NLS-En4DVar algorithm, it is further reduced into POD-4DVar (Tian et al. 2008; Tian et al. 2011), which is a typical 4DEnVar method without iteration.

The remainder of this paper is organized as follows. In section 2, we first describe the alluded to NLS reformulation of the standard En4DVar and the Gauss–Newton algorithm. We then present several simplified algorithms of the full Gauss–Newton algorithm and examine their pros and cons. In section 3, we carry out several numerical experiments using the Lorenz-96 model to evaluate the performance of all the proposed algorithms with a focus on their assimilation accuracy and computational complexity. We then perform a group of evaluation experiments using the Advanced Research WRF (ARW) Model to further exploit the potential applicability of three members of the NLS_i-En4DVar family of methods to real atmosphere and/or ocean models. Finally, we finish the paper with a summary and a few concluding remarks in section 4.

2. Methodology

The incremental form of 4DVar is defined as seeking the analysis increment for the initial conditions (ICs), $\mathbf{x}'_a$ (at the initial time t_0), such that $\mathbf{x}'_a$ minimizes the following cost function:

$$J(\mathbf{x}')=\frac{1}{2}(\mathbf{x}')^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}')+\frac{1}{2}\sum_{k=0}^{S}[H_kM_{t_0\to t_k}(\mathbf{x}_b+\mathbf{x}')-\mathbf{y}_{\mathrm{obs},k}]^{\mathrm{T}} \\ \times \mathbf{R}_k^{-1}[H_kM_{t_0\to t_k}(\mathbf{x}_b+\mathbf{x}')-\mathbf{y}_{\mathrm{obs},k}]. \tag{1a}$$

It is easy to check that Eq. (1a) can be rewritten as

$$J(\mathbf{x}') = \frac{1}{2}(\mathbf{x}')^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}') + \frac{1}{2}\sum_{k=0}^{S}[L'_k(\mathbf{x}') - \mathbf{y}'_{\mathrm{obs},k}]^{\mathrm{T}} \times \mathbf{R}_k^{-1}[L'_k(\mathbf{x}') - \mathbf{y}'_{\mathrm{obs},k}], \tag{1b}$$

where $\mathbf{x}' \in \mathbb{R}^{m_x}$ (m_x is the dimension of the state vector) is the perturbation of the background field $\mathbf{x}_b$ at t_0 and

$$L'_k(\mathbf{x}') = L_k(\mathbf{x}_b + \mathbf{x}') - L_k(\mathbf{x}_b), \tag{2}$$

$$\mathbf{y}'_{\mathrm{obs},k} = \mathbf{y}_{\mathrm{obs},k} - L_k(\mathbf{x}_b), \tag{3}$$

$$L_k = H_k M_{t_0 \to t_k}. \tag{4}$$

Here, the superscript T stands for the matrix transpose, $\mathbf{y}_{\mathrm{obs},k} \in \mathbb{R}^{m_{y,k}}$ is the observation at time incidence t_k, $m_{y,k}$ is the dimension of the observational vector $\mathbf{y}_{\mathrm{obs},k}$, b is the background value, $S+1$ is the total number of observation time points in the assimilation window, H_k is the observation operator, and $\mathbf{B}$ and $\mathbf{R}_k$ are the background and observation error covariance matrices, respectively. All the symbols used in this paper follow the general conventions and are described in Table 1.

In En4DVar, the background error covariance matrix $\mathbf{B}$ is approximated by the ensemble covariance matrix $\mathbf{B}_e$ as

$$\mathbf{B} \approx \mathbf{B}_e = \frac{(\mathbf{P}_x)(\mathbf{P}_x)^{\mathrm{T}}}{N-1}, \tag{5}$$

where $\mathbf{P}_x = (\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_N)$ is the prepared initial perturbations (IPs) and N is the ensemble size.

To filter out the noise resulting from undersampling (Houtekamer and Mitchell 2001), a decomposed correlation function operator (Liu et al. 2009; Buehner 2005) is often applied to modify the IP matrix $\mathbf{P}_x$ as follows:

$$\mathbf{P}_{x,\rho} = (\mathbf{P}_x < e > \boldsymbol{\rho}_x) = (\boldsymbol{\rho}_x \circ \mathbf{P}^*_{x,1}, \ldots, \boldsymbol{\rho}_x \circ \mathbf{P}^*_{x,N}) \tag{6a}$$

and thus

$$\mathbf{B} \approx \mathbf{B}_{e,\rho} = \mathbf{B}_e \circ \boldsymbol{\rho} = \frac{1}{N-1}\boldsymbol{\rho} \circ (\mathbf{P}_x\mathbf{P}_x^{\mathrm{T}}) = \frac{1}{N-1}\mathbf{P}_{x,\rho}(\mathbf{P}_{x,\rho}^{\mathrm{T}}), \tag{6b}$$

where $\mathbf{P}_{x,\rho} \in \mathbb{R}^{m_x \times (r \times N)}$ [r is the number of eigenvectors chosen; see Liu et al. (2009)] is the localized IP matrix, $\mathbf{P}^*_{x,j}$ ($j = 1, \ldots, N$) is an $m_x \times r$ matrix whose every column is the jth column of $\mathbf{P}_x$, $\boldsymbol{\rho}_x \in \mathbb{R}^{m_x \times r}$, and $\boldsymbol{\rho} = \boldsymbol{\rho}_x\boldsymbol{\rho}_x^{\mathrm{T}}$. Here, $\boldsymbol{\rho} \in \mathbb{R}^{m_x \times m_x}$ is the spatial correlation matrix (Tian et al. 2016; Zhang and Tian 2017, manuscript submitted to *J. Geophys. Res. Atmos.*, hereafter ZT) and $\mathbf{B} \circ \mathbf{C}$ stands for the Schür product of matrices $\mathbf{B}$ and $\mathbf{C}$, which is a matrix whose (i, j) entry is given by $b_{i,j} \times c_{i,j}$. In real data assimilation problems, the direct decomposition of $\boldsymbol{\rho}$ ($=\boldsymbol{\rho}_x\boldsymbol{\rho}_x^{\mathrm{T}}$) may not be feasible because of its huge size. ZT proposed an efficient local correlation matrix decomposition approach to compute the decomposition $\boldsymbol{\rho} = \boldsymbol{\rho}_x\boldsymbol{\rho}_x^{\mathrm{T}}$. For ease of the subsequent presentation, we define the operator $<e>$ in Eq. (6a).

To avoid the direct computation of the inverse of matrix $\mathbf{B}_{e,\rho}$, we further express the analysis solution $\mathbf{x}'_a$ by the linear combinations of $\mathbf{P}_{x,\rho}$ as

$$\mathbf{x}'_a = \mathbf{P}_{x,\rho}\boldsymbol{\beta}. \tag{7}$$

Substituting Eqs. (6) and (7) into Eq. (1) and expressing the cost function in terms of the new control variable $\boldsymbol{\beta}$ yields

$$\begin{aligned} J(\boldsymbol{\beta}) &= \frac{1}{2}(\mathbf{P}_{x,\rho}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{B}_{e,\rho})^{-1}(\mathbf{P}_{x,\rho}\boldsymbol{\beta}) + \frac{1}{2}\sum_{k=0}^{S}[L'_k(\mathbf{P}_{x,\rho}\boldsymbol{\beta}) - \mathbf{y}'_{\mathrm{obs},k}]^{\mathrm{T}}\mathbf{R}_k^{-1}[L'_k(\mathbf{P}_x\boldsymbol{\beta}) - \mathbf{y}'_{\mathrm{obs},k}] \\ &= \frac{1}{2}(N-1) \times \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta} + \frac{1}{2}\sum_{k=0}^{S}[L'_k(\mathbf{P}_{x,\rho}\boldsymbol{\beta}) - \mathbf{y}'_{\mathrm{obs},k}]^{\mathrm{T}}\mathbf{R}_k^{-1}[L'_k(\mathbf{P}_{x,\rho}\boldsymbol{\beta}) - \mathbf{y}'_{\mathrm{obs},k}]. \end{aligned} \tag{8}$$

Obviously, because of the nonlinearity, the solution to Eq. (8) can be computed iteratively by performing evaluations of the cost function Eq. (8) and its gradient,

$$\nabla J(\boldsymbol{\beta}) = (N-1)\boldsymbol{\beta} + \sum_{k=0}^{S}\mathbf{P}_{x,\rho}^{\mathrm{T}}(\mathbf{M}'_k)^{\mathrm{T}}(\mathbf{H}'_k)^{\mathrm{T}} \times \mathbf{R}_k^{-1}[L'_k(\mathbf{P}_{x,\rho}\boldsymbol{\beta}) - \mathbf{y}'_{\mathrm{obs},k}], \tag{9}$$

using a suitable descent algorithm [e.g., the limited-memory quasi-Newton method (L-BFGS) method; Liu and Nocedal (1989)]. Here, $\mathbf{M}'_k$ and $\mathbf{H}'_k$ are the tangent linear (TL) models of $M_{t_0 \to t_k}$ and H_k, respectively. Their respective adjoint models are $(\mathbf{M}'_k)^{\mathrm{T}}$ and $(\mathbf{H}'_k)^{\mathrm{T}}$. It is not hard to see that the adjoint models $(\mathbf{M}'_k)^{\mathrm{T}}$ and $(\mathbf{H}'_k)^{\mathrm{T}}$ are indispensable to the En4DVar problem given by Eqs. (8) and (9).

TABLE 1. Summary of key notation used in this article, where m_x is the size of the state space, $m_{y,k}$ is the number of observations at t_k, $S+1$ is the number of time steps, r is the number of eigenvectors chosen in the EOF decomposition of correlation matrix $\boldsymbol{\rho}$, and N is the number of ensemble members.

Notation	Description	Size
$\mathbf{x}_b, \mathbf{x}_a, \mathbf{x}', \mathbf{x}'_a, \mathbf{x}'_j$	State vector: time t_0 [b, background; a, analysis; prime ($'$), increment; j, ensemble member]	m_x
$\mathbf{P}_x$	$(\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_N)$	$m_x \times N$
$\mathbf{B}$	Background error covariance	$m_x \times m_x,$
$\mathbf{R}_k$	Observation error covariance	$m_{y,k} \times m_{y,k}$
$M_{t_0 \to t_k}(\cdot)$	Forecast model: time $t_0 \to t_k$	m_x (in, out)
$\mathbf{M}'_k$	Tangent linear forecast model: time $t_{k-1} \to t_k$	m_x (in, out)
$H_k(\cdot)$	Nonlinear observation operator: time t_k	m_x (in), $m_{y,k}$ (out)
$\mathbf{H}'_k$	Tangent linear observation model: time t_k	m_x (in), $m_{y,k}$ (out)
$L_k(\cdot)$	$H_k M_{t_0 \to t_k}(\cdot)$: time $t_0 \to t_k$	m_x (in), $m_{y,k}$ (out)
$\mathbf{y}_{\mathrm{obs},k}$	Observations: time t_k	$m_{y,k}$
$L'_k(\mathbf{x}')$	$L_k(\mathbf{x}_b + \mathbf{x}') – L_k(\mathbf{x}_b)$: time $t_0 \to t_k$	m_x (in), $m_{y,k}$ (out)
$\mathbf{y}'_{\mathrm{obs},k}$	$\mathbf{y}_{\mathrm{obs},k} - L_k(\mathbf{x}_b)$	$m_{y,k}$
$\boldsymbol{\rho}$	Spatial correlation matrix	$m_x \times m_x$
$\boldsymbol{\rho}_x$	$\boldsymbol{\rho}_x \boldsymbol{\rho}_x^{\mathrm{T}} = \boldsymbol{\rho}$	$m_x \times r$
$\mathbf{P}^{*}_{x,j}$	Every column is the jth column of $\mathbf{P}_x$	$m_x \times r$
$\mathbf{P}_{x,\rho}$	$\mathbf{P}_x < e > \boldsymbol{\rho}_x$	$m_x \times (r \times N)$
$\boldsymbol{\beta}^i$	Coefficient vector at the ith iteration	$(r \times N)$
$\mathbf{P}^{i-1}_{y,k}$	$\mathbf{H}'_k \mathbf{M}'^{,i-1}_k \mathbf{P}_x$: $(i-1)$th iteration	$m_{y,k} \times N$
$\mathbf{y}'^{a}_{k,j}$	$L_k(\mathbf{x}^{i-1}_a + \mathbf{x}'_j) – L_k(\mathbf{x}^{i-1}_a)$: $(i-1)$th iteration	$m_{y,k}$
$\mathbf{P}^{a}_{y,k}$	$(\mathbf{y}'^{a}_{k,1}, \ldots, \mathbf{y}'^{a}_{k,N})$	$m_{y,k} \times N$
$\mathbf{y}'_{k,j}$	$L_k(\mathbf{x}_b + \mathbf{x}'_j) – L_k(\mathbf{x}_b)$	$m_{y,k}$
$\mathbf{P}_{y,k}$	$(\mathbf{y}'_{k,1}, \ldots, \mathbf{y}'_{k,N})$	$m_{y,k} \times N$
$\mathbf{P}^{*,i-1}_{y,k}$	$\left\{ -(N-1)\left[\sum_{k=0}^{S} (\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}^{i-1}_{y,k}) + (N-1)\mathbf{I}\right]^{-1} \left[\sum_{k=0}^{S} (\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}} (\mathbf{P}^{i-1}_{y,k})\right]^{-1} (\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}} \right\}^{\mathrm{T}}$	$m_{y,k} \times N$
$\mathbf{P}^{\#,i-1}_{y,k}$	$\left\{ \left[\sum_{k=0}^{S} (\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}^{i-1}_{y,k}) + (N-1)\mathbf{I}\right]^{-1} (\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}} \right\}^{\mathrm{T}}$	$m_{y,k} \times N$
$\mathbf{P}^{*,a}_{y,k}$	$\left\{ -(N-1)\left[\sum_{k=0}^{S} (\mathbf{P}^{a}_{y,k})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}^{a}_{y,k}) + (N-1)\mathbf{I}\right]^{-1} \left[\sum_{k=0}^{S} (\mathbf{P}^{a}_{y,k})^{\mathrm{T}} (\mathbf{P}^{a}_{y,k})\right]^{-1} (\mathbf{P}^{a}_{y,k})^{\mathrm{T}} \right\}^{\mathrm{T}}$	$m_{y,k} \times N$
$\mathbf{P}^{\#,a}_{y,k}$	$\left\{ \left[\sum_{k=0}^{S} (\mathbf{P}^{a}_{y,k})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}^{a}_{y,k}) + (N-1)\mathbf{I}\right]^{-1} (\mathbf{P}^{a}_{y,k})^{\mathrm{T}} \right\}^{\mathrm{T}}$	$m_{y,k} \times N$
$\mathbf{P}^{*}_{y,k}$	$\left\{ -(N-1)\left[\sum_{k=0}^{S} (\mathbf{P}_{y,k})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}_{y,k}) + (N-1)\mathbf{I}\right]^{-1} \left[\sum_{k=0}^{S} (\mathbf{P}_{y,k})^{\mathrm{T}} (\mathbf{P}_{y,k})\right]^{-1} (\mathbf{P}_{y,k})^{\mathrm{T}} \right\}^{\mathrm{T}}$	$m_{y,k} \times N$
$\mathbf{P}^{\#}_{y,k}$	$\left\{ \left[\sum_{k=0}^{S} (\mathbf{P}_{y,k})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}_{y,k}) + (N-1)\mathbf{I}\right]^{-1} (\mathbf{P}_{y,k})^{\mathrm{T}} \right\}^{\mathrm{T}}$	$m_{y,k} \times N$
$\mathbf{P}^{*}_{y}$	$\{-(N-1)[(\mathbf{P}_y)^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{P}_y) + (N-1)\mathbf{I}]^{-1} [(\mathbf{P}_y)^{\mathrm{T}} (\mathbf{P}_y)]^{-1} (\mathbf{P}_y)^{\mathrm{T}}\}^{\mathrm{T}}$	$m_y \times N$
$\mathbf{P}^{\#}_{y}$	$\{[(\mathbf{P}_y)^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{P}_y) + (N-1)\mathbf{I}]^{-1} (\mathbf{P}_y)^{\mathrm{T}}\}^{\mathrm{T}}$	$m_y \times N$

a. *NLS$_0$-En4DVar: Gauss–Newton solution of En4DVar*

To circumvent the use of the adjoint models, we follow Tian and Feng (2015) by first rewriting Eq. (8) as a nonlinear least squares problem (Dennis and Schnabel 1996), which minimizes the following quadratic cost function:

$$J(\boldsymbol{\beta}) = \frac{1}{2} Q(\boldsymbol{\beta})^{\mathrm{T}} Q(\boldsymbol{\beta}), \qquad (10)$$

where

$$Q(\boldsymbol{\beta}) = \left\{ \begin{matrix} \sqrt{N-1}\boldsymbol{\beta} \\ \mathbf{R}^{-1/2}_{+,0}[L'_0(\mathbf{P}_{x,\rho}\boldsymbol{\beta}) - \mathbf{y}'_{\mathrm{obs},0}] \\ \vdots \\ \mathbf{R}^{-1/2}_{+,S}[L'_S(\mathbf{P}_{x,\rho}\boldsymbol{\beta}) - \mathbf{y}'_{\mathrm{obs},S}] \end{matrix} \right\} \qquad (11)$$

and $(\mathbf{R}^{1/2}_{+,k})(\mathbf{R}^{1/2}_{+,k})^{\mathrm{T}} = \mathbf{R}_k$. Consequently, the first-derivative matrix (or Jacobian matrix) $J_{\mathrm{ac}} Q(\boldsymbol{\beta})$ of $Q(\boldsymbol{\beta})$ can be computed as follows:

$$J_{\mathrm{ac}}Q(\boldsymbol{\beta}) = \frac{\partial Q(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \begin{pmatrix} \sqrt{N-1}\mathbf{I} \\ \mathbf{R}_{+,0}^{-1/2}\mathbf{H}_0'\mathbf{M}_0'\mathbf{P}_{x,\rho} \\ \vdots \\ \mathbf{R}_{+,S}^{-1/2}\mathbf{H}_S'\mathbf{M}_S'\mathbf{P}_{x,\rho} \end{pmatrix}, \tag{12}$$

where $\mathbf{I}$ denotes the $(N \times r) \times (N \times r)$ identity matrix. Therefore, the Gauss–Newton iteration for the nonlinear least squares problem Eq. (10) is defined by (Dennis and Schnabel 1996)

$$\boldsymbol{\beta}^i = \boldsymbol{\beta}^{i-1} - \{[J_{\mathrm{ac}}Q(\boldsymbol{\beta}^{i-1})]^{\mathrm{T}} J_{\mathrm{ac}}Q(\boldsymbol{\beta}^{i-1})\}^{-1} \times [J_{\mathrm{ac}}Q(\boldsymbol{\beta}^{i-1})]^{\mathrm{T}} Q(\boldsymbol{\beta}^{i-1}) \tag{13}$$

for $i = 1, \ldots, I_{\max}$, where $I_{\max}$ is the maximum iteration number.

Substituting Eqs. (11) and (12) into Eq. (13), we obtain

$$\boldsymbol{\beta}^i = \boldsymbol{\beta}^{i-1} - \left[(N-1)\mathbf{I} + \sum_{k=0}^{S} (\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})\right]^{-1} \times \left\{\sum_{k=0}^{S} (\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})^{\mathrm{T}}\mathbf{R}_k^{-1}[L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1}) - \mathbf{y}_{\mathrm{obs},k}'] + (N-1)\boldsymbol{\beta}^{i-1}\right\} \tag{14a}$$

and thus

$$\left[(N-1)\mathbf{I} + \sum_{k=0}^{S} (\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})\right]\Delta\boldsymbol{\beta}^i = -\left\{\sum_{k=0}^{S} (\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})^{\mathrm{T}}\mathbf{R}_k^{-1}[L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1}) - \mathbf{y}_{\mathrm{obs},k}'] + (N-1)\boldsymbol{\beta}^{i-1}\right\}, \tag{14b}$$

where $\Delta\boldsymbol{\beta}^i = \boldsymbol{\beta}^i - \boldsymbol{\beta}^{i-1}$. Consequently, we arrive at $\boldsymbol{\beta}^i = \boldsymbol{\beta}^{i-1} + \Delta\boldsymbol{\beta}^i$ and then $\mathbf{x}_a^{i,\prime} = \mathbf{P}_{x,\rho}\boldsymbol{\beta}^i$ by solving the linear system Eq. (14b) using the conjugate gradient (CG) method or a preconditioned CG method.

We name the iterative algorithm in Eq. (14) NLS$_0$-En4D-Var since it is the solution to the nonlinear least squares problem in Eq. (10). Compared with the original En4DVar iterative algorithm in Eqs. (8) and (9), the formulation of NLS$_0$-En4DVar is derived from Eq. (8) through Eqs. (10)–(14) without making any additional assumptions. So the subscript 0 is used to indicate that the formulation Eq. (14) is derived from Eqs. (8) and (9) without any simplification; thus, Eq. (14) is mathematically equivalent to Eqs. (8) and (9), namely, NLS$_0$-En4DVar is equivalent to the original En4DVar. However, the adjoint models are nicely avoided in NLS$_0$-En4DVar Eq. (14) since $(\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})^{\mathrm{T}}$ can be obtained simply by transposing $(\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})$, which can be formed explicitly by integrating the tangent linear models. Furthermore, Eq. (14) is easier to implement and more portable because it does not rely on any existing optimization package (Tian and Feng 2017). One may argue that the computational cost may be significantly increased because the ensemble size is expanded to $N \times r$ from N after applying the localization [i.e. Eq. (6)]. However, the situation may not be as bad as first imagined in view of the rapid development of high-performance computers. The tangent model M_k^{i-1} has two basic input variables: $\mathbf{x}_{i-1}$ and $\mathbf{x}'$. In the ith iteration step, $\mathbf{x}_{i-1}$ is assigned the value $\mathbf{x}_a^{i-1}$, which is used to initialize the forecast model $M_{t_0 \to t_k}$ to prepare the parameters for the tangent model $\mathbf{M}_k'^{,i-1}$. After that, all the parameters of the tangent model are fixed during the ith iterative step (Zou et al. 1997). That means the computation of $\mathbf{M}_k'^{,i-1}\mathbf{P}_x$ only needs to call the forecast model $M_{t_0 \to t_k}$ once, which is followed by $N \times r$ series of linear operations in $\mathbf{M}_k'^{,i-1}$. In addition, the implementation of $N \times r$ series of linear operations in $\mathbf{M}_k'^{,i-1}$ can be carried out in parallel. The situation is further improved in the following simplified version of NLS$_0$-En4DVar.

b. NLS$_1$-En4DVar: Simplification by acting the localization in observation space

In view of the possible high computational cost of evaluating $\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho}$, we further approximate $(\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_{x,\rho})$ by $(\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_x)_\rho$ in NLS$_0$-En4DVar, which then leads to NLS$_1$-En4DVar, which is defined as

$$\left[(N-1)\mathbf{I} + \sum_{k=0}^{S} (\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_x)_\rho^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_x)_\rho\right]\Delta\boldsymbol{\beta}^i = -\left\{\sum_{k=0}^{S} (\mathbf{H}_k'\mathbf{M}_k'^{,i-1}\mathbf{P}_x)_\rho^{\mathrm{T}}\mathbf{R}_k^{-1}[L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1}) - \mathbf{y}_{\mathrm{obs},k}'] + (N-1)\boldsymbol{\beta}^{i-1}\right\}, \tag{15}$$

where $(\mathbf{H}'_k\mathbf{M}'^{,i-1}_k\mathbf{P}_x)_\rho = (\mathbf{P}^{i-1}_{y,k} < e > \boldsymbol{\rho}_y)$, $\mathbf{P}^{i-1}_{y,k} = \mathbf{H}'_k\mathbf{M}'^{,i-1}_k\mathbf{P}_x$, and $\boldsymbol{\rho}_y \in \mathbb{R}^{m_y \times r}$ should be computed altogether with $\boldsymbol{\rho}_x$ (ZT). The following approximation

$$(\mathbf{H}'_k\mathbf{M}'^{,i-1}_k\mathbf{P}_{x,\rho}) \approx (\mathbf{H}'_k\mathbf{M}'^{,i-1}_k\mathbf{P}_x)_\rho \tag{16}$$

is adopted to alleviate the computational cost because only N series of linear operations in $\mathbf{M}'^{,i-1}_k$ are required for computing $\mathbf{H}'_k\mathbf{M}'^{,i-1}_k\mathbf{P}_x$, which is considerably less than the original $N \times r$ series of linear operations in $\mathbf{M}'^{,i-1}_k$ for computing $\mathbf{H}'_k\mathbf{M}'^{,i-1}_k\mathbf{P}_{x,\rho}$ in NLS$_0$-En4DVar.

It should be noted that solving (15) by using the CG method is still extremely challenging since the size $(N \times r)$ of $\boldsymbol{\beta}^i$ is extremely large ($\sim 10^{6-9}$) as a result of the localization implementation [Eqs. (6) and (16)]. To avoid the expensive computational cost resulting from the CG iterations, we first go back to the unlocalized version of Eq. (15):

$$\Delta\boldsymbol{\beta}^i = -\left[(N-1)\mathbf{I} + \sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{P}^{i-1}_{y,k})\right]^{-1} \times \left\{\sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}[L'_k(\mathbf{P}_x\boldsymbol{\beta}^{i-1}) - \mathbf{y}'_{\mathrm{obs},k}] + (N-1)\boldsymbol{\beta}^{i-1}\right\}, \tag{15$'$}$$

and then transform it into following format:

$$\Delta\boldsymbol{\beta}^i = -\left[(N-1)\mathbf{I} + \sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{P}^{i-1}_{y,k})\right]^{-1}\left\{\sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}[L'_k(\mathbf{P}_x\boldsymbol{\beta}^{i-1}) - \mathbf{y}'_{\mathrm{obs},k}] + (N-1)\left[\sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}(\mathbf{P}^{i-1}_{y,k})\right]^{-1}\sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}\mathbf{H}'_k\mathbf{M}'^{,i-1}_k(\mathbf{P}_x\boldsymbol{\beta}^{i-1})\right\}. \tag{15$''$}$$

Equation (15″) is derived by substituting $\boldsymbol{\beta}^{i-1} \approx [(\mathbf{P}^{i-1}_y)^{\mathrm{T}}(\mathbf{P}^{i-1}_y)]^{-1}(\mathbf{P}^{i-1}_y)^{\mathrm{T}}\mathbf{H}'\mathbf{M}'^{,i-1}(\mathbf{P}_x\boldsymbol{\beta}^{i-1})$ into the term $(N-1)\boldsymbol{\beta}^{i-1}$ on the right side of Eq. (15′), where the derivation of $\boldsymbol{\beta}^{i-1} \approx [(\mathbf{P}^{i-1}_y)^{\mathrm{T}}(\mathbf{P}^{i-1}_y)]^{-1}(\mathbf{P}^{i-1}_y)^{\mathrm{T}}\mathbf{H}'\mathbf{M}'^{,i-1}(\mathbf{P}_x\boldsymbol{\beta}^{i-1})$ is based on the assumption $\mathbf{H}'\mathbf{M}'^{,i-1}(\mathbf{P}_x\boldsymbol{\beta}^{i-1}) \approx \mathbf{P}^{i-1}_y\boldsymbol{\beta}^{i-1}$, where

$$\mathbf{H}'\mathbf{M}'^{,i-1} = \begin{pmatrix} \mathbf{H}'_0\mathbf{M}'^{,i-1}_0 \\ \mathbf{H}'_1\mathbf{M}'^{,i-1}_1 \\ \vdots \\ \mathbf{H}'_S\mathbf{M}'^{,i-1}_S \end{pmatrix} \quad \text{and}$$

$$\mathbf{P}^{i-1}_y = \begin{pmatrix} \mathbf{P}^{i-1}_{y,0} \\ \mathbf{P}^{i-1}_{y,1} \\ \vdots \\ \mathbf{P}^{i-1}_{y,S} \end{pmatrix}.$$

We further mark

$$(\mathbf{P}^{*,i-1}_{y,k})^{\mathrm{T}} = -(N-1)\left[\sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{P}^{i-1}_{y,k}) + (N-1)\mathbf{I}\right]^{-1} \times \left[\sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}(\mathbf{P}^{i-1}_{y,k})\right]^{-1}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}, \tag{16a}$$

and

$$(\mathbf{P}^{\#,i-1}_{y,k})^{\mathrm{T}} = -\left[\sum_{k=0}^{S}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{P}^{i-1}_{y,k}) + (N-1)\mathbf{I}\right]^{-1}(\mathbf{P}^{i-1}_{y,k})^{\mathrm{T}}. \tag{16b}$$

Therefore, Eq. (15″) is further transformed as follows:

$$\Delta\boldsymbol{\beta}^i = \sum_{k=0}^{S}(\mathbf{P}^{\#,i-1}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}[L'_k(\mathbf{P}_x\boldsymbol{\beta}^{i-1}) - \mathbf{y}'_{\mathrm{obs},k}] + \sum_{k=0}^{S}(\mathbf{P}^{*,i-1}_{y,k})^{\mathrm{T}}\mathbf{H}'_k\mathbf{M}'^{,i-1}_k(\mathbf{P}_x\boldsymbol{\beta}^{i-1}). \tag{15$'''$}$$

Subsequently, we adapt the alternative localization scheme proposed by Tian and Feng (2015) to transform Eq. (15‴) into the following format,

$$\Delta\boldsymbol{\beta}^i = \sum_{k=0}^{S}(\mathbf{P}^{\#,i-1}_{y,k} < e > \boldsymbol{\rho}_y)^{\mathrm{T}}\mathbf{R}_k^{-1}[L'_k(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1}) - \mathbf{y}'_{\mathrm{obs},k}] + \sum_{k=0}^{S}(\mathbf{P}^{*,i-1}_{y,k} < e > \boldsymbol{\rho}_y)^{\mathrm{T}}\mathbf{H}'_k\mathbf{M}'^{,i-1}_k(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1}), \tag{17}$$

by localizing the matrices $\mathbf{P}^{\#,i-1}_{y,k}$, $\mathbf{P}^{*,i-1}_{y,k}$, and $\mathbf{P}_x$ through $(\mathbf{P}^{*,i-1}_{y,k} < e > \boldsymbol{\rho}_y)$, $(\mathbf{P}^{\#,i-1}_{y,k} < e > \boldsymbol{\rho}_y)$, and $\mathbf{P}_{x,\rho} = (\mathbf{P}_x < e > \boldsymbol{\rho}_x)$. Apparently, $\Delta\boldsymbol{\beta}^i$ is now computed explicitly by Eq. (17) and the CG iterations are thus avoided. By direct calculation, we obtain

$$\Delta\mathbf{x}'^{i}_a = \mathbf{P}_{x,\rho}\Delta\boldsymbol{\beta}^i$$

$$
\begin{aligned}
&=\mathbf{P}_{x,\rho}\sum_{k=0}^{S}[(\mathbf{P}_{y,k}^{*,i-1}<e>\boldsymbol{\rho}_y)^{\mathrm{T}}\mathbf{H}_k'\mathbf{M}_k'^{,i-1}(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})]\\
&\quad+\mathbf{P}_{x,\rho}\sum_{k=0}^{S}(\mathbf{P}_{y,k}^{\#,i-1}<e>\boldsymbol{\rho}_y)^{\mathrm{T}}\mathbf{R}_k^{-1}[\mathbf{y}_{\mathrm{obs},k}'-L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})]\\
&=\sum_{k=0}^{S}\{[\boldsymbol{\rho}_x\boldsymbol{\rho}_y^{\mathrm{T}}\circ\mathbf{P}_x(\mathbf{P}_{y,k}^{*,i-1})^{\mathrm{T}}]\mathbf{H}_k'\mathbf{M}_k'^{,i-1}(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})\}\\
&\quad+\sum_{k=0}^{S}[\boldsymbol{\rho}_x\boldsymbol{\rho}_y^{\mathrm{T}}\circ\mathbf{P}_x(\mathbf{P}_{y,k}^{\#,i-1})^{\mathrm{T}}]\mathbf{R}_k^{-1}[\mathbf{y}_{\mathrm{obs},k}'-L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})]\\
&=\sum_{k=0}^{S}\{[\boldsymbol{\rho}_{xy}\circ\mathbf{P}_x(\mathbf{P}_{y,k}^{*,i-1})^{\mathrm{T}}]\mathbf{H}_k'\mathbf{M}_k'^{,i-1}(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})\}\\
&\quad+\sum_{k=0}^{S}[\boldsymbol{\rho}_{xy}\circ\mathbf{P}_x(\mathbf{P}_{y,k}^{\#,i-1})^{\mathrm{T}}]\mathbf{R}_k^{-1}[\mathbf{y}_{\mathrm{obs},k}'-L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})],
\end{aligned}
\tag{18}
$$

which shows that the localization scheme adopted in Eq.(17) is completely equivalent to (but with higher computational efficiency than) that proposed by Tian and Feng (2015). In fact, the last line in Eq. (18) has exactly the same localization form as Eq. (30b) in Tian and Feng (2015). Here, $\boldsymbol{\rho}_{xy}$ $(=\boldsymbol{\rho}_x\boldsymbol{\rho}_y^{\mathrm{T}})$ is the correlation matrix between model states and observational variables (ZT).

c. NLS$_2$-En4DVar: Further simplification by abandoning the use of the tangent models

In NLS$_1$-En4DVar, the tangent models $\mathbf{M}_k'^{,i-1}$ and $\mathbf{H}_k'$ are still retained and their implementation still could be a bit difficult despite being much easier compared to the implementation of their adjoint models. To further reduce the implementation complexity by abandoning the use of the tangent models, we introduce NLS$_2$-En4DVar, which is formulated as follows:

$$
\begin{aligned}
\Delta\boldsymbol{\beta}^i=&\sum_{k=0}^{S}[(\mathbf{P}_{y,k}^{*,a}<e>\boldsymbol{\rho}_y)^{\mathrm{T}}L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})]\\
&+\sum_{k=0}^{S}(\mathbf{P}_{y,k}^{\#,a}<e>\boldsymbol{\rho}_y)^{\mathrm{T}}\mathbf{R}_k^{-1}[\mathbf{y}_{\mathrm{obs},k}'-L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})],
\end{aligned}
\tag{19a}
$$

where

$$
\begin{aligned}
(\mathbf{P}_{y,k}^{*,a})^{\mathrm{T}}=&-(N-1)\left[\sum_{k=0}^{S}(\mathbf{P}_{y,k}^{a})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{P}_{y,k}^{a})+(N-1)\mathbf{I}\right]^{-1}\\
&\times\left[\sum_{k=0}^{S}(\mathbf{P}_{y,k}^{a})^{\mathrm{T}}(\mathbf{P}_{y,k}^{a})\right]^{-1}(\mathbf{P}_{y,k}^{a})^{\mathrm{T}},
\end{aligned}
\tag{19b}
$$

$$
(\mathbf{P}_{y,k}^{\#,a})^{\mathrm{T}}=\left[\sum_{k=0}^{S}(\mathbf{P}_{y,k}^{a})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{P}_{y,k}^{a})+(N-1)\mathbf{I}\right]^{-1}(\mathbf{P}_{y,k}^{a})^{\mathrm{T}},
\tag{19c}
$$

$$
\mathbf{P}_{y,k}^{a}=(\mathbf{y}_{k,1}'^{a},\ldots,\mathbf{y}_{k,N}'^{a}),
\tag{19d}
$$

and

$$
\mathbf{y}_{k,j}'^{a}=L_k(\mathbf{x}_a^{i-1}+\mathbf{x}_j')-L_k(\mathbf{x}_a^{i-1}),j=1,\ldots,N.
\tag{19e}
$$

In NLS$_2$-En4DVar, the forecast model $M_{t_0\rightarrow t_k}$ will be repeatedly integrated N times at the ith iteration to form $\mathbf{P}_{y,k}^{a}$ by means of $\mathbf{P}_{y,k}^{a}=L_k(\mathbf{x}_a^{i-1}+\mathbf{P}_x)-L_k(\mathbf{x}_a^{i-1})$, which certainly leads to expensive computational costs.

d. NLS$_3$-En4DVar: Further simplification by fixing the OPs

To reduce the computational costs, we fix the simulated OPs, $\mathbf{P}_{y,k}^{a}=\mathbf{P}_{y,k}$, during the iteration process and thus form the following NLS$_3$-En4DVar method:

$$
\begin{aligned}
\Delta\boldsymbol{\beta}^i=&\sum_{k=0}^{S}[(\mathbf{P}_{y,k}^{*}<e>\boldsymbol{\rho}_y)^{\mathrm{T}}L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})]\\
&+\sum_{k=0}^{S}(\mathbf{P}_{y,k}^{\#}<e>\boldsymbol{\rho}_y)^{\mathrm{T}}\mathbf{R}_k^{-1}[\mathbf{y}_{\mathrm{obs},k}'-L_k'(\mathbf{P}_{x,\rho}\boldsymbol{\beta}^{i-1})],
\end{aligned}
\tag{20a}
$$

where

$$
\begin{aligned}
(\mathbf{P}_{y,k}^{*})^{\mathrm{T}}=&-(N-1)\left[\sum_{k=0}^{S}(\mathbf{P}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{P}_{y,k})+(N-1)\mathbf{I}\right]^{-1}\\
&\times\left[\sum_{k=0}^{S}(\mathbf{P}_{y,k})^{\mathrm{T}}(\mathbf{P}_{y,k})\right]^{-1}(\mathbf{P}_{y,k})^{\mathrm{T}},
\end{aligned}
\tag{20b}
$$

$$
(\mathbf{P}_{y,k}^{\#})^{\mathrm{T}}=\left[\sum_{k=0}^{S}(\mathbf{P}_{y,k})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{P}_{y,k})+(N-1)\mathbf{I}\right]^{-1}(\mathbf{P}_{y,k})^{\mathrm{T}},
\tag{20c}
$$

$$
\mathbf{P}_{y,k}=(\mathbf{y}_{k,1}',\ldots,\mathbf{y}_{k,N}'),
\tag{20d}
$$

and

$$
\mathbf{y}_{k,j}'=L_k(\mathbf{x}_b+\mathbf{x}_j')-L_k(\mathbf{x}_b).
\tag{20e}
$$

It is interesting to note that NLS$_3$-En4DVar is just the NLS-4DVar method (but with a modified localization scheme) proposed by Tian and Feng (2015).

e. NLS$_4$-En4DVar: One iteration of NLS$_3$-En4DVar

Performing only one iteration of NLS$_3$-En4DVar Eq. (20) with $\boldsymbol{\beta}^0=0$, we obtain the following NLS$_4$-En4DVar method:

$$\boldsymbol{\beta}^1 = \sum_{k=0}^{S} (\mathbf{P}^{\#}_{y,k} < e > \boldsymbol{\rho}_y)^{\mathrm{T}} \mathbf{R}_k^{-1} \mathbf{y}'_{\mathrm{obs},k}, \qquad (21)$$

which is just the POD-4DVar method proposed by Tian et al. (2011). Moreover, restricting Eq. (21) to the 3D case, we obtain

$$\boldsymbol{\beta}^1 = (\mathbf{P}^{\#}_y < e > \boldsymbol{\rho}_y)^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{y}'_{\mathrm{obs}} \qquad (22a)$$

and thus

$$\mathbf{x}'_a = \mathbf{P}_{x,\rho} (\mathbf{P}^{\#}_y < e > \boldsymbol{\rho}_y)^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{y}'_{\mathrm{obs}}, \qquad (22b)$$

where

$$(\mathbf{P}^{\#}_y)^{\mathrm{T}} = [(\mathbf{P}_y)^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{P}_y) + (N-1)\mathbf{I}]^{-1} (\mathbf{P}_y)^{\mathrm{T}} \qquad (22c)$$

and

$$\mathbf{P}_y = H(\mathbf{x}_b + \mathbf{P}_x) - H(\mathbf{x}_b), \qquad (22d)$$

which is essentially the same as the LETKF method (Hunt et al. 2007); they differ only in their localization schemes. Here, H and $\mathbf{R}$ are the observation operator and observation error covariance, respectively, for the 3D case.

f. NLS$_5$-En4DVar: An iterative improvement scheme

Following Courtier et al. (1994), a linear approximation to Eq. (1a) is proposed, which is valid in the vicinity of $\mathbf{x}_b$, as follows:

$$\begin{cases} J(\delta\mathbf{x}_0) = \dfrac{1}{2}\{(\delta\mathbf{x}_0)^{\mathrm{T}} \mathbf{B}^{-1} (\delta\mathbf{x}_0) + \displaystyle\sum_{k=0}^{S} [\Upsilon_k \delta\mathbf{x}_0 + L_k(\mathbf{x}_b) - \mathbf{y}_{\mathrm{obs},k}]^{\mathrm{T}} \\ \qquad \times \mathbf{R}_k^{-1} [\Upsilon_k \delta\mathbf{x}_0 + L_k(\mathbf{x}_b) - \mathbf{y}_{\mathrm{obs},k}]\} \\ \Upsilon_k = \mathbf{H}'_k \mathbf{M}'_k \\ \delta\mathbf{x}_0^a = \mathrm{Argmin}\, J(\delta\mathbf{x}_0) \\ \mathbf{x}_a = \mathbf{x}_b + \delta\mathbf{x}_0^a \end{cases} \qquad (23)$$

Equation (23) is widely taken as the inner loop, in which the reduced resolution tangent linear and adjoint models with some simplified physical processes are the common choice of many operational centers. To account for some nonlinearities, an outer loop is added and the concept of guess field $\mathbf{x}_a^i$ is introduced as follows:

$$\begin{cases} J_i(\delta\mathbf{x}_0^{i-1}) = \dfrac{1}{2}[(\delta\mathbf{x}_0^{i-1} + \mathbf{x}_a^{\prime,i-1})^{\mathrm{T}} \mathbf{B}^{-1} (\delta\mathbf{x}_0^{i-1} + \mathbf{x}_a^{\prime,i-1}) \\ \quad + \displaystyle\sum_{k=0}^{S} (\Upsilon_k \delta\mathbf{x}_0^{i-1} - \mathbf{y}_{\mathrm{obs},k}^{\prime,i-1})^{\mathrm{T}} \mathbf{R}_k^{-1} (\Upsilon_k \delta\mathbf{x}_0^{i-1} - \mathbf{y}_{\mathrm{obs},k}^{\prime,i-1})] \\ \mathbf{x}_a^{\prime,i-1} = \mathbf{x}_a^{i-1} - \mathbf{x}_b \\ \mathbf{y}_{\mathrm{obs},k}^{\prime,i-1} = \mathbf{y}_{\mathrm{obs},k} - L_k(\mathbf{x}_a^{i-1}) \\ \Upsilon_k = \mathbf{H}'_k \mathbf{M}'_k \\ \delta\mathbf{x}_0^{i-1,*} = \mathrm{Arg\,min} J_i(\delta\mathbf{x}_0^{i-1}) \\ \mathbf{x}_a^i = \mathbf{x}_a^{i-1} + \delta\mathbf{x}_0^{i-1,*} \end{cases} \qquad (24)$$

for $i = 1, \ldots, I_{\max}$, where $\mathbf{x}_a^0 = \mathbf{x}_b$. Thus, we propose an iterative improvement method, called NLS$_5$-En4DVar, of the NLS$_i$-En4DVar ($i = 0, 1, 2, 3$) family. The NLS$_5$ -En4DVar approach also assumes that $\delta\mathbf{x}_0^{i-1}$ is expressed by the linear combinations of the IPs $\mathbf{P}_x^{i-1} = (\mathbf{x}_1^{\prime,i-1}, \ldots, \mathbf{x}_N^{\prime,i-1})$ as follows:

$$\delta\mathbf{x}_0^{i-1} = \mathbf{P}_x^{i-1} \delta\boldsymbol{\beta}^{i-1}. \qquad (25)$$

Substituting Eq. (25), $\mathbf{x}_a^{\prime,i-1} = \mathbf{P}_x^{i-1} \boldsymbol{\beta}^{i-1}$, and the ensemble background covariance, $\mathbf{B} = [(\mathbf{P}_x^{i-1})(\mathbf{P}_x^{i-1})^{\mathrm{T}}]/(N-1)$, into Eq. (24) and expressing the cost function in terms of the new control variable $\delta\boldsymbol{\beta}^{i-1}$ yields

$$\begin{aligned} J_i(\delta\boldsymbol{\beta}^{i-1}) = \frac{1}{2}&(N-1)(\delta\boldsymbol{\beta}^{i-1} + \boldsymbol{\beta}^{i-1})^{\mathrm{T}} (\delta\boldsymbol{\beta}^{i-1} + \boldsymbol{\beta}^{i-1}) \\ &+ \sum_{k=0}^{S} (\Upsilon_k \mathbf{P}_x^{i-1} \delta\boldsymbol{\beta}^{i-1} - \mathbf{y}_{\mathrm{obs},k}^{\prime,i-1})^{\mathrm{T}} \\ &\times \mathbf{R}_k^{-1} (\Upsilon_k \mathbf{P}_x^{i-1} \delta\boldsymbol{\beta}^{i-1} - \mathbf{y}_{\mathrm{obs},k}^{\prime,i-1})\Big], \end{aligned} \qquad (26)$$

where $\boldsymbol{\beta}^{i-1} = [(\mathbf{P}_x^{i-1})^{\mathrm{T}} \mathbf{P}_x^{i-1}]^{-1} (\mathbf{P}_x^{i-1})^{\mathrm{T}} \mathbf{x}_a^{\prime,i-1}$. Following the idea of the NLS$_4$-En4DVar approach, we can obtain

$$\begin{aligned} \delta\boldsymbol{\beta}^{i-1} &= \left[(N-1)\mathbf{I} + \sum_{k=0}^{S} (\mathbf{P}_{y,k}^{i-1})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}_{y,k}^{i-1})\right]^{-1} \left[\sum_{k=0}^{S} (\mathbf{P}_{y,k}^{i-1})^{\mathrm{T}} \mathbf{R}_k^{-1} \mathbf{y}_{\mathrm{obs},k}^{\prime,i-1} - (N-1)\boldsymbol{\beta}^{i-1}\right] \\ &= \left[(N-1)\mathbf{I} + \sum_{k=0}^{S} (\mathbf{P}_{y,k}^{i-1})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}_{y,k}^{i-1})\right]^{-1} \times \left\{\sum_{k=0}^{S} (\mathbf{P}_{y,k}^{i-1})^{\mathrm{T}} \mathbf{R}_k^{-1} \mathbf{y}_{\mathrm{obs},k}^{\prime,i-1} - (N-1)[(\mathbf{P}_x^{i-1})^{\mathrm{T}} \mathbf{P}_x^{i-1}]^{-1} (\mathbf{P}_x^{i})^{\mathrm{T}} \mathbf{x}_a^{\prime,i-1}\right\}, \end{aligned} \qquad (27)$$

where $\mathbf{P}_{y,k}^{i-1} = (\mathbf{y}_{k,1}^{\prime,i-1}, \ldots, \mathbf{y}_{k,N}^{\prime,i-1}) = \Upsilon_k \mathbf{P}_x^{i-1}$. After the localization process, we can further obtain

$$\delta\boldsymbol{\beta}_\rho^{i-1} = \sum_{k=0}^{S} (\mathbf{P}_{y,k}^{\#,i-1} < e > \boldsymbol{\rho}_y)^{\mathrm{T}} \mathbf{R}_k^{-1} \mathbf{y}_{\mathrm{obs},k}^{\prime,i-1} + (\mathbf{P}_x^{\#,i-1} < e > \boldsymbol{\rho}_x)^{\mathrm{T}} \mathbf{x}_a^{\prime,i-1}, \quad (28a)$$

$$\delta\mathbf{x}_0^{i-1,*} = \mathbf{P}_{x,\rho}^{i-1} \delta\boldsymbol{\beta}_\rho^{i-1}, \quad (28b)$$

and

$$\mathbf{x}_a^i = \mathbf{x}_a^{i-1} + \delta\mathbf{x}_0^{i-1,*}, \quad (28c)$$

where

$$(\mathbf{P}_{y,k}^{\#,i-1})^{\mathrm{T}} = \left[(N-1)\mathbf{I} + \sum_{k=0}^{S} (\mathbf{P}_{y,k}^{i-1})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}_{y,k}^{i-1})\right]^{-1} (\mathbf{P}_{y,k}^{i-1})^{\mathrm{T}} \quad (29a)$$

and

$$(\mathbf{P}_x^{\#,i-1})^{\mathrm{T}} = -(N-1)\left[(N-1)\mathbf{I} + \sum_{k=0}^{S} (\mathbf{P}_{y,k}^{i-1})^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{P}_{y,k}^{i-1})\right]^{-1} \times [(\mathbf{P}_x^{i-1})^{\mathrm{T}} \mathbf{P}_x^{i-1}]^{-1} (\mathbf{P}_x^{i-1})^{\mathrm{T}}. \quad (29b)$$

Obviously, we can fix $\mathbf{P}_x^{i-1} = \mathbf{P}_x = (\mathbf{x}_1', \mathbf{x}_2', \ldots, \mathbf{x}_N')$ during the outer loop. In this case, we have to update their corresponding OPs $\mathbf{P}_{y,k}^{i-1} = (\mathbf{y}_{k,1}^{\prime,i-1}, \ldots, \mathbf{y}_{k,N}^{\prime,i-1})$ through $\mathbf{y}_{k,j}^{\prime,i-1} = L_k(\mathbf{x}_a^{i-1} + \mathbf{x}_j') - L_k(\mathbf{x}_a^{i-1})$, which should run the forecast model N times repeatedly for each outer loop and thus lead to expensive computational costs. To address this, we can let

$$\mathbf{P}_x^{i-1} = (\mathbf{x}_1^{\prime,i-1}, \ldots, \mathbf{x}_N^{\prime,i-1}) \quad \text{and} \quad (30a)$$

$$\mathbf{x}_j^{\prime,i-1} = \mathbf{x}_j' + \mathbf{x}_b - \mathbf{x}_a^{i-1}, \quad (30b)$$

and then

$$\mathbf{y}_{k,j}^{\prime,i-1} = L_k(\mathbf{x}_a^{i-1} + \mathbf{x}_j^{\prime,i-1}) - L_k(\mathbf{x}_a^{i-1}) = L_k(\mathbf{x}_b + \mathbf{x}_j') - L_k(\mathbf{x}_a^{i-1}), \quad (30c)$$

which means that one only needs to produce the ensemble model integrations [i.e., $L_k(\mathbf{x}_b + \mathbf{x}_j')$, $j = 1, \ldots, N$] once, which could be repeatedly utilized during the outer loop in the NLS$_5$-En4DVar. The advantage of this strategy is that we can approximate $\mathbf{H}_k'\mathbf{M}_k'\mathbf{P}_x^{i-1}$ at $\mathbf{x}_a^{i-1}$ without running the forecast model N times repeatedly.

We note that the above NLS$_5$-En4DVar method takes the NLS$_4$-En4DVar analysis solution as the starting value $\mathbf{x}_a^1$ and improves it through an iterative procedure. Each iteration of NLS$_5$-En4DVar consists of three steps. First, the background field $\mathbf{x}_a^{i-1}$ is updated; that in turn calls for updates of $\mathbf{P}_x^{i-1}$ and $\mathbf{P}_{y,k}^{i-1}$ in the second step. Finally, an incremental $\delta\boldsymbol{\beta}_\rho^{i-1}$ is computed by calling a quasi-NLS$_4$-En4DVar scheme. Repeating this process then results in NLS$_5$-En4DVar. Compared to the iterative methods NLS$_i$-En4DVar ($i = 0, \ldots, 3$), NLS$_5$-En4DVar has one important advantage: it does not involve the tangent model, nor does it use the (expensive) adjoint model while it can update $\mathbf{P}_{y,k}^{i-1}$ with $\mathbf{x}_a^{i-1}$ during the iteration process. As a result, it should be cheaper to implement this approach while achieving almost the same accuracy as NLS$_1$-En4DVar, as demonstrated by the numerical tests performed in section 3a.

To summarize, NLS$_0$-En4DVar provides an alternative approach (i.e., a Gauss–Newton iterative method) to the En4DVar formulation Eq. (8) without dealing with the adjoint models. This new approach is based on the idea of first transforming the original En4DVar formulation into a nonlinear least squares problem, which then is solved by a Gauss–Newton iterative algorithm. To reduce the computation complexity of the algorithm, we propose a series of simplified methods, namely, NLS$_i$-En4DVar ($i = 0, \ldots, 4$), with decreasing computational complexity at the expense of reduced accuracy. It is interesting to note that these NLS$_i$-En4DVar ($i = 0, \ldots, 4$) methods unify the representation of En4DVar, 4DEnVar, and LETKF into a single formulation and clearly reveal the relationship between these methods. They also provide a better understanding of these methods from a different perspective and more flexible choices for data assimilation for different situations. Furthermore, we also propose an improved iterative method (i.e., NLS$_5$-En4DVar) based on the comprehensive analysis on the above NLS$_i$-En4DVar family of methods. Particularly, an efficient localization scheme is introduced into the NLS$_i$-En4DVar ($i = 1, \ldots, 5$) methods to reduce the expensive computational costs resulting from the CG iterations required for solving Eq. (15) in the original NLS$_1$-En4DVar formulations.

3. Preliminary numerical evaluations

We design two groups of numerical experiments to evaluate the proposed NLS$_i$-En4DVar methods. The first group of experiments is based on the Lorenz-96 model, and they are designed to better understand their inherent relationship. The second group of experiments is based on the Advanced Research WRF (ARW) Model, and they are used to further explore the potential applicability of three members of the

NLS$_i$-En4DVar family of methods to real atmosphere and/or ocean models.

a. *Preliminary evaluations using the Lorenz-96 model*

In this section, we use the Lorenz-96 model (Lorenz 1996) as the model platform to evaluate all the proposed methods (i.e., NLS$_i$-En4DVar, $i = 0, 1, 2, 3, 4, 5$ and En4DVar). This model has been widely used to study various issues associated with data assimilation. For example, it has been used for the comparative study of 4DVar, 4DEnKF (Fertig et al. 2007), and EnKF (Kalnay et al. 2007), as well as for evaluations of EnKF without perturbed observations (Whitaker and Hamill 2002), 4DEnKF (Hunt et al. 2004), and 4DEnVar (Tian et al. 2011). Therefore, we choose the Lorenz-96 model to investigate the advantages and disadvantages of the seven data assimilation methods discussed in the previous section.

The Lorenz-96 model is governed by the following system of nonlinear equations:

$$\frac{dx_i}{dt} = -x_{i-2}x_{i-1} + x_{i-1}x_{i+1} - x_i + F, i = 1, \ldots, n, \qquad (31)$$

with periodic boundary conditions. The model behaves quite differently with different values of F and produces chaotic systems with integer values of F larger than three. In this configuration, we take $n = 40$ and $F = 8$. For computational stability, a time step of 0.05 units (or 6 h in equivalent; Lorenz 1996) is adopted and a fourth-order Runge–Kutta scheme is used for temporal integration in this study. In all cases, the true states and observations were generated by the model with $F = 8$. These observations were then assimilated into models with $F = 8$ and 9. The default number of observations is $m = 20$ (equally spaced at every observation time). Observations were taken every two time steps (or 12 h), which were generated by adding random error perturbations with the standard deviation error of 0.1 to the time series of the true state. All experiments were carried out over 5 days. The default parameter setups are the ensemble size $N = 30$, $r = 10$, and the covariance localization Schür radius $r_0 = d_{h,0}(d_{v0}) = 8$ (only one direction in the Lorenz-96 model). The length of the assimilation window is four steps and the ensemble is regenerated by the LETKF format (Hunt et al. 2004) for each cycle. The background initial field is generated by integrating the model 100 000 steps from an arbitrary nonzero field and the true initial field is subsequently obtained by integrating the 20 steps from the background initial field. Therefore, this background state is significantly different from the "true" state (not shown). The initial sample is generated by running the Lorenz model from 90 steps (i.e., sampling once every 3 time steps).

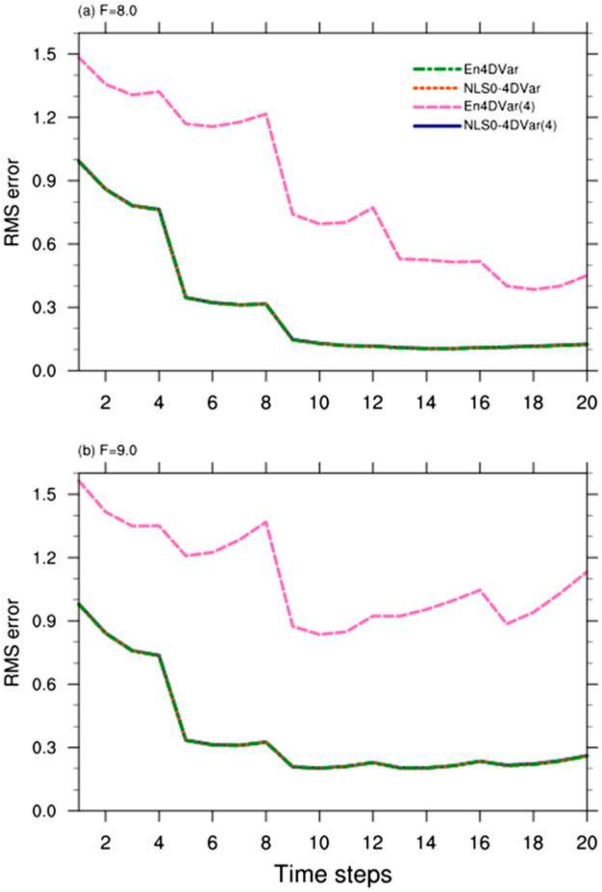

FIG. 1. Time series of the RMSEs for the four data assimilation techniques (En4DVar, NLS$_0$-En4DVar, En4DVar with $I_{\max} = 4$, and NLS$_0$-En4DVar with $I_{\max} = 4$) using default parameter setups with the Lorenz-96 model (a) without model error ($F = 8$) and (b) with model error ($F = 9$).

The performance of the NLS$_0$-En4DVar approach is first examined in comparison to the original En4DVar method. In this study, we employ an L-BFGS method (Liu and Nocedal 1989) for the minimization in the En4DVar approach through Eqs. (8) and (9). Correspondingly, the Gauss–Newton algorithm with the conjugate gradient linear solver is used to solve NLS$_0$-En4DVar. We compare the performance of NLS$_0$-En4DVar with the original En4DVar method under the perfect-model (Fig. 1a: $F = 8$ for all the truth, forecast, and assimilation runs) and the imperfect-model (Fig. 1b: $F = 8$ for the truth run and $F = 9$ for the forecast and assimilation runs) assumptions. The results show that both methods perform equally satisfactorily in terms of overall low root-mean-square

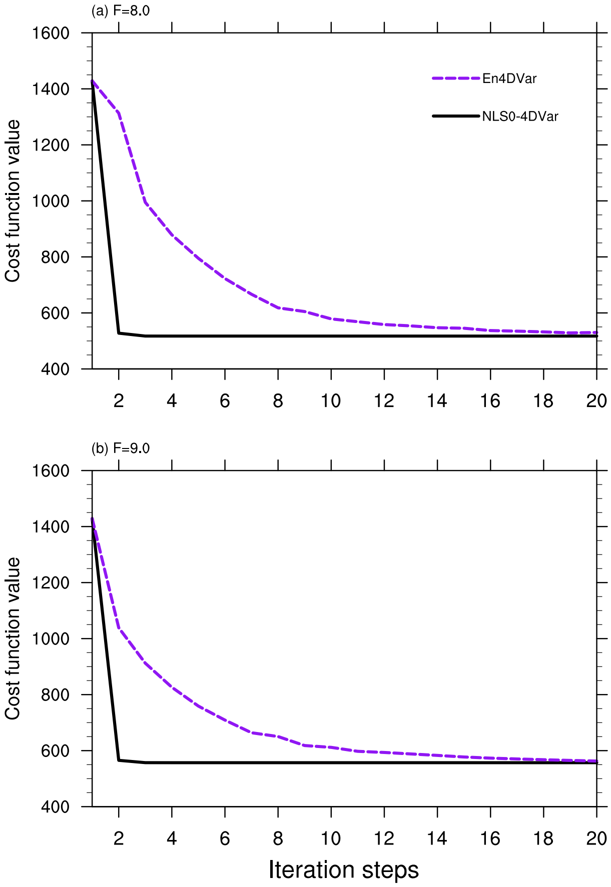

FIG. 2. Shown are the variations of the cost function values with increasing iteration steps for the two data assimilation techniques (En4DVar and NLS$_0$-En4DVar) using default parameter setups with the Lorenz-96 model (a) without model error ($F = 8$) and (b) with model error ($F = 9$).

(RMS) errors. As noted above, NLS$_0$-En4DVar is just a reformulation of the En4DVar method, so they are mathematically equivalent. The difference between the two assimilation methods is the minimization algorithms (the L-BFGS versus the Gauss–Newton algorithm) used inside both methods. Numerically, these two methods should differ on the assimilation performance because the adjoint models must be involved in En4DVar but not in NLS$_0$-En4DVar.

To further examine the differences between the two iterative algorithms (L-BFGS and Gauss–Newton), we focus on their performance in the first assimilation window, and their convergence speeds in particular are examined since their RMS errors are almost identical. Figure 2 shows that the cost function values are decreasing in the iteration steps (only the first 20 steps) for the two methods under both the perfect and imperfect model assumptions. In the perfect model case, the Gauss–Newton algorithm reaches its minimum (517.156) after 5 iterations, while the L-BFGS used in the original En4DVar method needs 82 iterations to reach its minimum value (which is also 517.156). Similarly, for the imperfect model, their minimum values (both are 556.94) are reached in 8 iterations by the Gauss–Newton algorithm for NLS$_0$-En4DVar and 83 iterations by the L-BFGS for the En4DVar method. To further check the influence of the iterations on the assimilation results, we also evaluate the performance of En4DVar and NLS$_0$-En4DVar with $I_{\max} = 4$ in Fig. 1. Obviously, the performance of En4DVar depends on the maximum iteration number and performs less satisfactory for smaller $I_{\max}$. As expected, the extra cost of integrating the adjoint models for 4DVar (due to the iterative algorithm based on the L-BFGS) seriously limits its applications for complicated real-world models. It should also be noted that NLS$_0$-En4DVar requires series of linear operations in $\mathbf{M}_k'^{,i-1}$, which certainly adds some computational costs, although the number of iterations is significantly smaller than that of the original En4DVar method.

Second, to investigate how the proposed simplifications affect the final analysis results, we also designed another group of experiments to test the NLS$_i$-En4DVar (i = 0, 1, 2, 3, 5) methods. As mentioned above, we observed that their assimilation accuracy is reduced gradually from NLS$_0$-En4DVar to NLS$_3$-En4DVar, but this transition features decreasing computational complexity, in both the perfect and imperfect-model scenarios (Fig. 3). By comparison, NLS$_i$-En4DVar ($i = 1, 2, 3$) performs only slightly to moderately worse than NLS$_0$-En4DVar does. This group of experiments illustrates that the tangent linear models (with the updated $\mathbf{x}_a^{i-1}$ during the iteration process) are vital to achieving higher assimilation accuracy. Remarkably, NLS$_5$-En4DVar performs almost as well as NLS$_1$-En4DVar and substantially better than NLS$_2$-En4DVar in the perfect-model scenario. Furthermore, it shows strong competition in the imperfect-model scenario that its performance is on a par with that of NLS$_0$-En4DVar [performs better than the NLS$_i$-En4DVar (i = 1, 2, 3) methods]. These results demonstrate that to use (or approximate) the tangent models with the updated $\mathbf{x}_a^i$ during the iteration process contributes noticeably to the accuracy of the final assimilation result.

Finally, to examine the sensitivity of the 4DEnVar assimilation performance to the iteration strategy, we also compared the performance of NLS$_3$-En4DVar (a typical 4DEnVar method) using different maximum iteration numbers ($I_{\max} = 1, 3, 5$). We note that NLS$_3$-En4DVar with $I_{\max} = 1$ (i.e., no iteration is performed) is just the NLS$_4$-En4DVar method. As expected, the performance of NLS$_3$-En4DVar becomes better and better

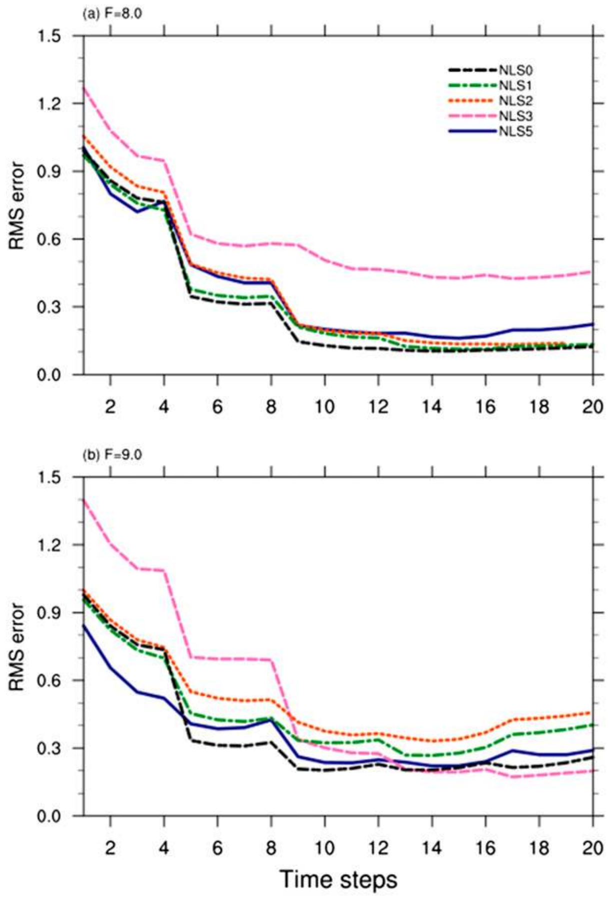

FIG. 3. Time series of the RMSEs for the five data assimilation techniques (NLS_0-En4DVar, NLS_1-En4DVar, NLS_2-En4DVar, NLS_3-En4DVar, and NLS_5-En4DVar) using default parameter setups with the Lorenz-96 model (a) without model error ($F = 8$) and (b) with model error ($F = 9$).

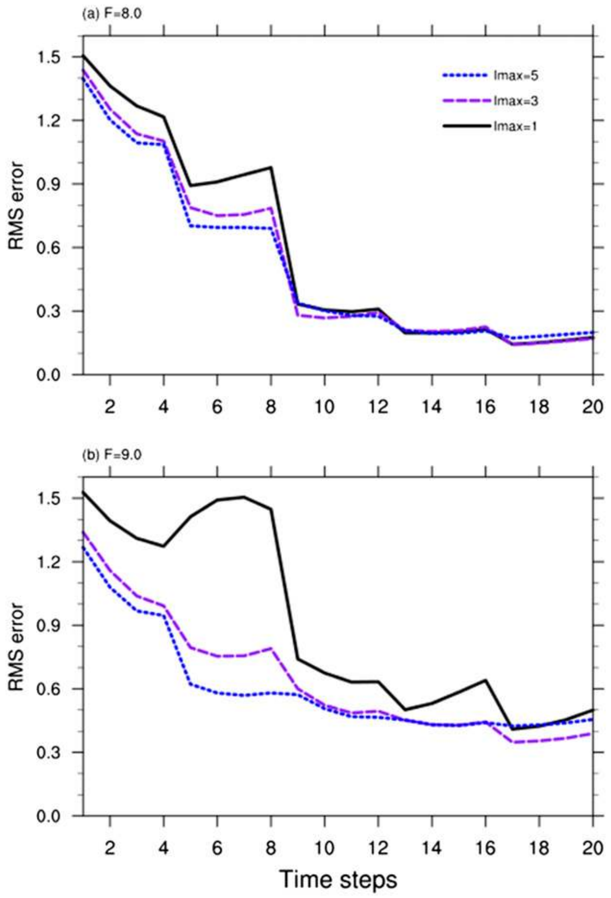

FIG. 4. Time series of the RMSEs for NLS_3-En4DVar with different maximum iteration numbers ($I_{max} = 1, 3, 5$) with the Lorenz-96 model (a) without model error ($F = 8$) and (b) with model error ($F = 9$).

as the iteration number increases (Fig. 4). Noticeably, NLS_4-En4DVar performs worse than the other methods. These results imply that the nonlinearity has a large impact on data assimilation and should not be ignored. The so-called no-iteration approach may not be a true advantage of an assimilation method because the small savings in computational cost are obtained at the expense of much reduced accuracy.

b. *Preliminary evaluations using the Advanced Research WRF (ARW) Model*

The new version ARW Model (ARW3.7) is used for the experiments of this section. All experiments are done using the mesh of 100 × 120 (latitude × longitude) grid points with grid spacing of 30 km. The domain, which covers the region of China from 15.5°–43.5°N to 88.5°–131.5°E, with 30 layers from $\eta = 0$ to $\eta = 1$ in the vertical direction, is used. The experiment design used here is almost same as that in Tian and Feng (2015). The NCEP Final (FNL) Operational Model Global Tropospheric Analyses (http://rda.ucar.edu/datasets/ds083.2/) are used as the first-guess field and boundary conditions in the experiments. The "true" and "background" (referred to as Ctrl) simulated fields and the synthetic observations are produced in the same way. The true and background fields are produced from 12- and 24-h forecasts initialized by the NCEP/FNL data at 12 and 24 h prior to the analysis time, respectively. We choose a 6-h assimilation window with accumulated rainfall and traditional temperature (only on the $\eta = 0.563$ layer) observation data available at 0, 3, and 6 h (at the observational stations; ZT) after the analysis time; the ensemble samples are also generated by the 4D moving sampling strategy (Tian and Feng 2015) and the ensemble size is 146. The horizontal localization radius used in this section is 750 km and $r = 42$ (ZT). The three NLS_i-En4DVar ($i = 3, 4, 5$) methods are tested to

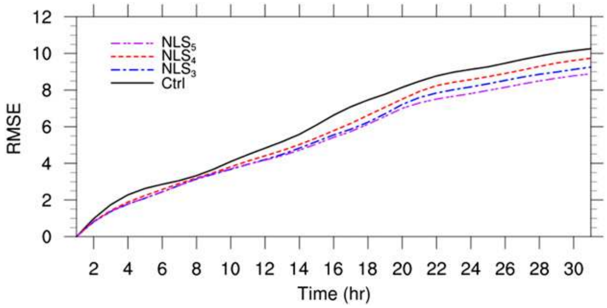

FIG. 5. RMSEs of 30-h forecasts of accumulated rainfall (from 1 to 30 h) with the ICs from the Ctrl, NLS$_3$, NLS$_4$, and NLS$_5$ methods.

examine their potential applicability to complicated real-world nonlinear weather and climate models. It should be noted here that the other three NLS$_i$-En4DVar ($i = 0, 1, 2$) methods are too difficult/costly to implement with the WRF Model. We set $I_{\max} = 3$ in these experiments to test the NLS$_3$-En4DVar and NLS$_5$-En4DVar methods.

To evaluate the overall performance of the NLS$_i$-En4DVar methods ($i = 3, 4, 5$), we first compare the root-mean-square errors (RMSEs) of 30-h forecasts of accumulated rainfall (1–30 h after the analysis time) with the ICs from the NLS$_i$ ($i = 3, 4, 5$) and Ctrl simulations, respectively. Figure 5 shows the RMSEs of 30-h forecasts of accumulated rainfall with the ICs from the assimilations; all three NLS$_i$-En4DVar methods produce smaller values than those from Ctrl, which indicates that the three NLS$_i$-En4DVar methods can steadily improve the rainfall forecast over 30 h. Moreover, Fig. 5 also shows that both NLS$_3$-En4DVar and NLS$_5$-En4DVar outperform NLS$_4$-En4DVar (i.e., POD-4DVar) moderately, which illustrates the importance of the iterative strategy adopted in the two NLS$_i$-En4DVar methods ($i = 3, 5$). In particular, the improved iterative method NLS$_5$-En4DVar performs slightly better than NLS$_3$-En4DVar in terms of its lower RMSEs of 30-h forecasts of accumulated rainfall.

Furthermore, to investigate the performance of the two NLS$_i$-En4DVar ($i = 3, 5$) methods, we also compare the vertical profiles of RMSEs of some basic model variables from NLS$_i$ ($i = 3, 5$) and Ctrl at the beginning and at the end of the assimilation window, which are calculated at all horizontal model grid points of each layer. The same conclusion can be drawn that NLS$_5$-En4DVar produces more accurate atmospheric states than NLS$_3$-En4DVar does. As seen in Figs. 6 and 7, NLS$_5$-En4DVar yields a smaller error than NLS$_3$-En4DVar does on most model layers. This further indicates that the improved iterative strategy adopted in NLS$_5$-En4DVar contributes in an impactful way to its superior performance.

Additionally, to examine the impacts of the modified localization implementation in Eq. (17) to the NLS$_i$-En4DVar methods, another version of NLS$_3$-En4DVar using the CG iterations [referred to as NLS$_3$-En4DVar-CG; Zhang et al. (2016)] is deliberately involved in the above comparisons. Generally, NLS$_3$-En4DVar performs almost completely the same as NLS$_3$-En4DVar-CG does in terms of the RMSEs (not shown), which illustrates that computational efficiency without the use of the CG iterations in NLS$_3$-En4DVar is not achieved at the expense of the assimilation accuracy. All of the above numerical experiments are conducted serially in a DELL M620 server with two CPUs and fourth-generation broadband cellular network (4G) memories.

The CPU times, which are required to accomplish their pure assimilation experiments (not including the forecast model run time), for the NLS$_i$-En4DVar ($i = 3, 4, 5$) and NLS$_3$-En4DVar-CG methods, are compared in Table 2. The results show that the improved iterative strategy adopted in NLS$_5$-En4DVar adds almost no (i.e., negligible) computational cost, which is also expected because the NLS$_5$-En4DVar reformulation does not add any additional forecast model runs. Furthermore, the comparison between the two versions of NLS$_3$-En4DVar illustrates that the introduction of the modified localization scheme without the CG iterations indeed results in a significant speed up. In addition, the CPU time for running the forecast model (i.e., the WRF model) once is about 10.3 min. We point out that the CPU times are always case and machine dependent.

4. Summary and conclusions

The only difference between En4DVar and the traditional 4DVar is that En4DVar uses the ensemble background covariance matrix $\mathbf{B}_e$ provided by the parallel EnKF implementation to replace the usual static, flow-independent covariance matrix $\mathbf{B}$ in the traditional 4DVar scheme. The inclusion of the matrix $\mathbf{B}_e$ enables (or facilitates) En4DVar to benefit from the flow-dependent strength of EnKF. However, the adjoint model is still required inside the minimization procedure in En4DVar. An alternative strategy is exploited in this study with a focus on circumventing the obstacle of the adjoint models. To accomplish this goal, we first transform the standard En4DVar into a nonlinear least squares problem. Subsequently, a Gauss–Newton iterative algorithm is proposed to iteratively approximate the optimal nonlinear 4DVar solution (without invoking the adjoint models), which defines NLS$_0$-En4DVar with no dependency on the adjoint models. To reduce the computational and implementation costs, the

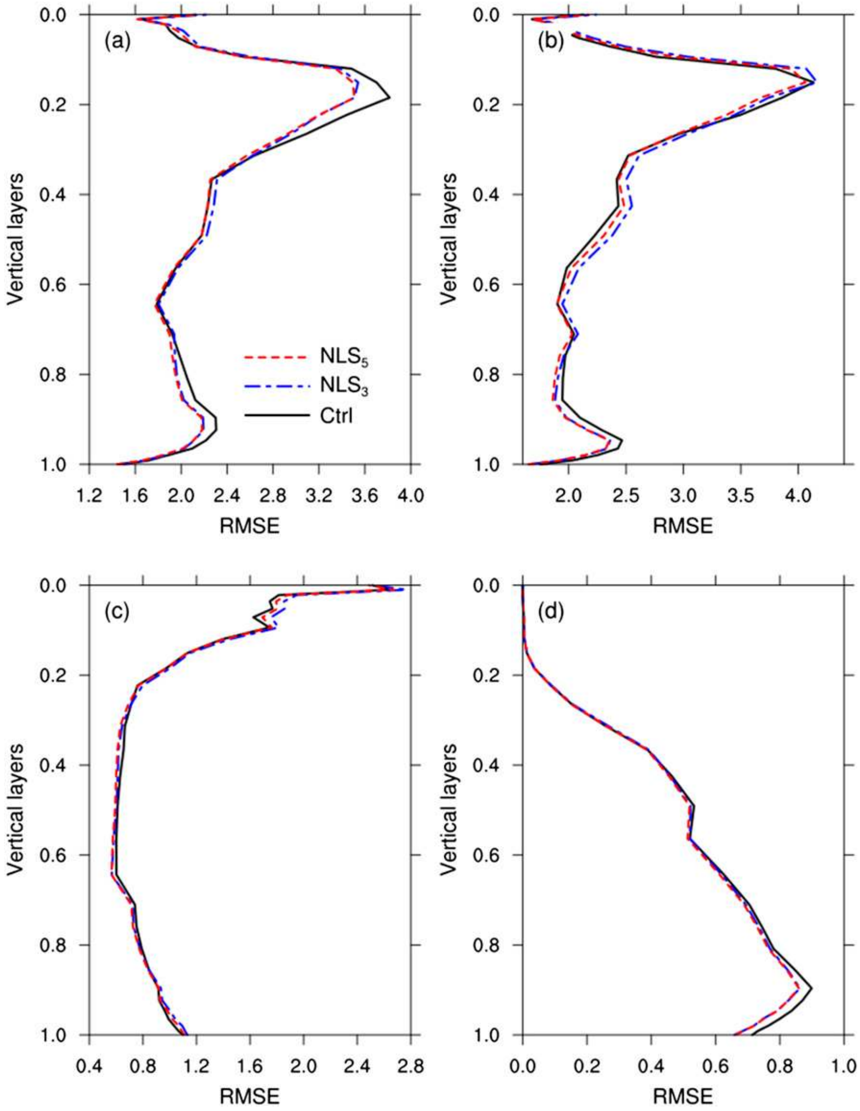

FIG. 6. Vertical profiles of RMSEs of (a) zonal wind (m s^{-1}), (b) meridional wind (m s^{-1}), (c) temperature (°C), and (d) water vapor mixing ratio (g kg^{-1}) of the NLS$_3$ and NLS$_5$ methods at the start of the assimilation window

Gauss–Newton iterative algorithm is further simplified with different levels of accuracy and computational complexity, which provides more flexible choices for data assimilation during different situations.

The advantages and disadvantages of the proposed NLS$_i$-En4DVar methods and their inherent relationship are demonstrated by several groups of numerical evaluation experiments with the Lorenz-96 and Advanced Research version of WRF (ARW) models, respectively. Our main conclusions are summarized as follows:

- The proposed NLS$_i$-En4DVar family of methods provides a unified formulation of En4DVar, 4DEnVar, and LETKF for better understanding their relationship and a new perspective for future research on data assimilation.
- NLS$_0$-En4DVar provides an alternative approach to the En4DVar approach by utilizing explicitly an iterative algorithm rather than resorting to any existing optimization algorithm. Unlike the usual En4DVar method, which involves performing several evaluations of the cost function and its gradient with the L-BFGS algorithm, the adjoint models are tactically avoided in the proposed Gauss–Newton algorithm. Moreover, the added computational cost could be compensated for by its ease of parallel computing.
- While looking forward in the future, NLS$_1$-En4DVar appears to be a very promising and competitive option

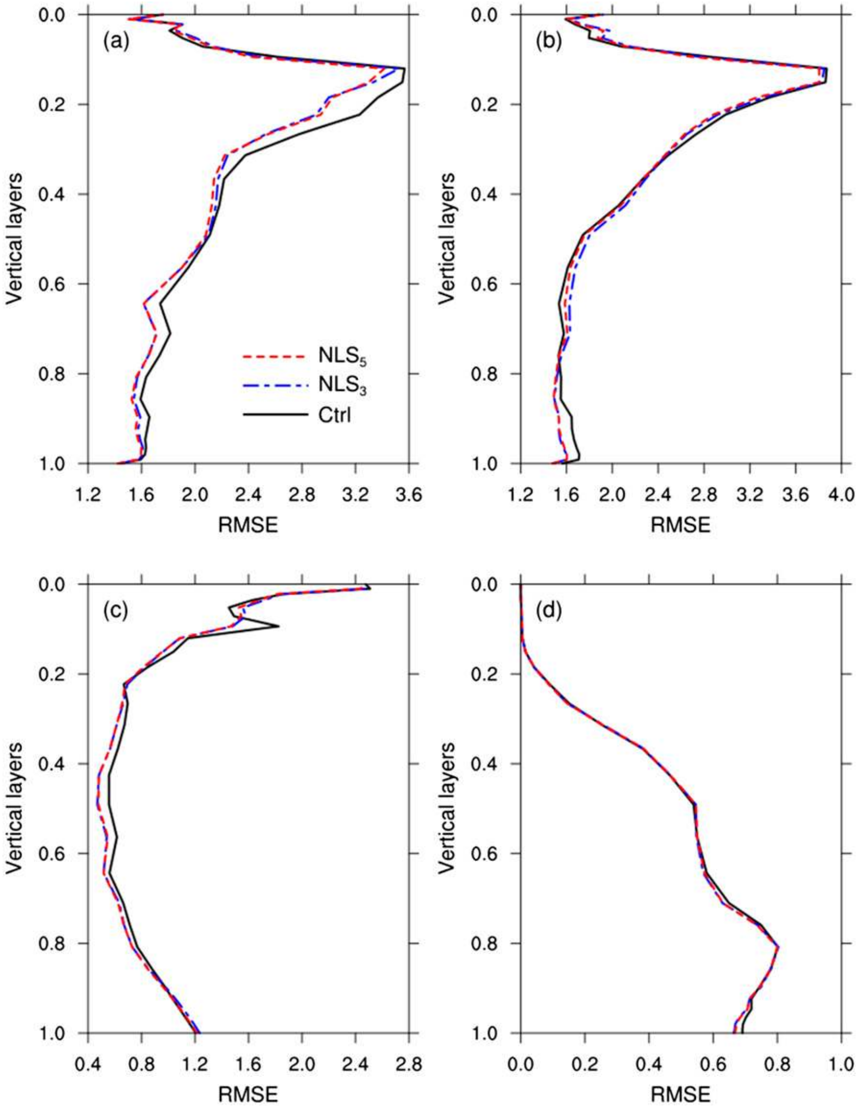

FIG. 7. As in Fig. 6, but at the end of the assimilation window.

because of its moderate difficulty in computer implementation (it only requires the tangent linear models as NLS$_0$-En4DVar does) and its significantly reduced computational costs, which all are achieved with only a small reduction in the convergence speed compared to the original NLS$_0$-En4DVar.

- Compared with NLS$_1$-En4DVar, NLS$_5$-En4DVar is quite advantageous and has great potential because of its ease of implementation and the absence of any need to invoke the tangent and adjoint models, as well as its superior performance.
- The iteration strategy adopted in the proposed NLS$_i$-En4DVar family of methods is expected to significantly improve the quality of the data assimilation results because of its ability to handle the strong nonlinearity of the models, which is common in data assimilation.

TABLE 2. The CPU time (s) needed for NLS$_i$-En4DVar ($i = 3, 4, 5$) and NLS$_3$-En4DVar-CG to accomplish their pure WRF assimilation experiments for each iteration.

Method	NLS$_3$-En4DVar	NLS$_3$-En4DVar-CG	NLS$_4$-En4DVar	NLS$_5$-En4DVar
CPU time (s)	258.72	222 102.73	86.24	258.82

Acknowledgments. We thank the three anonymous reviewers for their critical comments and suggestions, which helped to improve the manuscript. The work of XT was partially supported by the National Key Research and Development Program of China (2016YFA0600203), the High-resolution Earth Observation System Major Special Project (CHEOS; 32-Y20A17-9001-15/17), and the National Natural Science Foundation of China (41575100).

REFERENCES

Amezcua, J., M. Goodliff, and P. J. Van Leeuwen, 2017: A weak-constraint 4DEnsembleVar. Part I: Formulation and simple model experiments. *Tellus*, **69A**, 1271564, https://doi.org/10.1080/16000870.2016.1271564.

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

Bowler, N. E., and Coauthors, 2017a: Inflation and localization tests in the development of an ensemble of 4D-ensemble variational assimilations. *Quart. J. Roy. Meteor. Soc.*, **143**, 1280–1302, doi:10.1002/qj.3004.

——, and Coauthors, 2017b: The effect of improved ensemble covariances on hybrid variational data assimilation. *Quart. J. Roy. Meteor. Soc.*, **143**, 785–797, https://doi.org/10.1002/qj.2964.

Buehner, M., 2005: Ensemble derived stationary and flow dependent background error covariances: Evaluation in a quasi-operational NWP setting. *Quart. J. Roy. Meteor. Soc.*, **131**, 1013–1043, https://doi.org/10.1256/qj.04.15.

Clayton, A.M., A.C. Lorenc, and D.M. Barker, 2013: Operational implementation of a hybrid ensemble/4D-Var global data assimilation system at the Met Office. *Quart. J. Roy. Meteor. Soc.*, **139**, 1445–1461, doi:10.1002/qj.2054.

Courtier, P., J.-N. Thépaut, and A. Hollingsworth, 1994: A strategy for operational implementation of 4D-Var, using an incremental approach. *Quart. J. Roy. Meteor. Soc.*, **120**, 1367–1387, https://doi.org/10.1002/qj.49712051912.

Dennis, J. E., Jr., and R. B. Schnabel, 1996: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics, Vol. 16, SIAM, 378 pp.

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasigeostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, **99**, 10 143–10 162, https://doi.org/10.1029/94JC00572.

——, and P. J. Van Leeuwen, 2000: An ensemble Kalman smoother for nonlinear dynamics. *Mon. Wea. Rev.*, **128**, 1852–1867, https://doi.org/10.1175/1520-0493(2000)128<1852:AEKSFN>2.0.CO;2.

Fertig, E. J., J. Harlim, and B. R. Hunt, 2007: A comparative study of 4DVar and 4D ensemble Kalman filter: Perfect model simulations with Lorenz-96. *Tellus*, **59A**, 96–101, https://doi.org/10.1111/j.1600-0870.2006.00205.x.

Goodliff, M., J. Amezcua, and P. J. Van Leeuwen, 2017: A weak-constraint 4DEnsembleVar. Part II: Experiments with larger models. *Tellus*, **69A**, 1271565, https://doi.org/10.1080/16000870.2016.1271565.

Gustafsson, N., 2007: Discussion on '4D-Var or EnKF?' *Tellus*, **59A**, 774–777, https://doi.org/10.1111/j.1600-0870.2007.00262.x.

Houtekamer, P. L., and H. L. Mitchell, 2001: A sequential ensemble Kalman filter for atmospheric data assimilation. *Mon. Wea. Rev.*, **129**, 123–137, https://doi.org/10.1175/1520-0493(2001)129<0123:ASEKFF>2.0.CO;2.

——, B. He, and H. L. Mitchell, 2014: Parallel implementation of an ensemble Kalman filter. *Mon. Wea. Rev.*, **142**, 1163–1182, https://doi.org/10.1175/MWR-D-13-00011.1.

Hunt, B. R., and Coauthors, 2004: Four-dimensional ensemble Kalman filtering. *Tellus*, **56A**, 273–277, https://doi.org/10.3402/tellusa.v56i4.14424.

——, E. J. Kostelich, E. Ott, and I. Szunyogh, 2007: Efficient data assimilation spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D*, **230**, 112–126, https://doi.org/10.1016/j.physd.2006.11.008.

Kalnay, E., H. Li, T. Miyoshi, S.-C. Yang, and J. Ballabrera-Poy, 2007: 4D-Var or ensemble Kalman filter? *Tellus*, **59A**, 758–773, https://doi.org/10.1111/j.1600-0870.2007.00261.x.

Karmitsa, N., 2016: Numerical methods for large-scale nonsmooth optimization. *Big Data Optimization: Recent Developments and Challenges*, A. Emrouznejad, Ed., Studies in Big Data, Vol. 18, Springer, 419–436.

Liu, C., Q. Xiao, and B. Wang, 2008: An ensemble-based four-dimensional variational data assimilation scheme. Part I: Technique formulation and preliminary test. *Mon. Wea. Rev.*, **136**, 3363–3373, https://doi.org/10.1175/2008MWR2312.1.

——, ——, and ——, 2009: 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, https://doi.org/10.1175/2008MWR2699.1.

Liu, D. C., and J. Nocedal, 1989: On the limited memory BFGS method for large scale optimization. *Math. Program.*, **45**, 503–528, https://doi.org/10.1007/BF01589116.

Lorenc, A. C., 2003: The potential of the ensemble Kalman filter for NWP: A comparison with 4DVar. *Quart. J. Roy. Meteor. Soc.*, **129**, 3183–3203, https://doi.org/10.1256/qj.02.132.

——, 2013: Recommended nomenclature for EnVar data assimilation methods. Met Office, 2 pp., https://www.wcrp-climate.org/WGNE/BlueBook/2013/individual-articles/01_Lorenc_Andrew_EnVar_nomenclature.pdf.

——, 2017: Improving ensemble covariances in hybrid variational data assimilation without increasing ensemble size. *Quart. J. Roy. Meteor. Soc.*, **143**, 1062–1072, https://doi.org/10.1002/qj.2990.

——, M. Jardak, T. Payne, N. E. Bowler, and M. A. Wlasak, 2017: Computing an ensemble of variational data assimilations using its mean and perturbations. *Quart. J. Roy. Meteor. Soc.*, **143**, 798–805, https://doi.org/10.1002/qj.2965.

Lorenz, E., 1996: Predictability: A problem partly solved. *Proc. Seminar on Predictability*, Vol. 1, Reading, United Kingdom, ECMWF, 1–19.

Qiu, C., A. Shao, Q. Xu, and L. Wei, 2007: Fitting model fields to observations by using singular value decomposition: An ensemble-based 4DVar approach. *J. Geophys. Res.*, **112**, D11105, https://doi.org/10.1029/2006JD007994.

Rabier, F., H. Jarvinen, E. Klinker, J.-F. Mahfouf, and A. Simmons, 2000: The ECMWF operational implementation of four-dimensional variational physics. *Quart. J. Roy. Meteor. Soc.*, **126**, 1143–1170, https://doi.org/10.1002/qj.49712656415.

Steward, J. L., I. M. Navon, M. Zupanski, and N. Karmitsa, 2012: Impact of non-smooth observation operators on variational and sequential data assimilation for a limited-area shallow water equations model. *Quart. J. Roy. Meteor. Soc.*, **138**, 323–339, https://doi.org/10.1002/qj.935.

Tian, X. J., and Z. H. Xie, 2012: Implementations of a square-root ensemble analysis and a hybrid localization into the

POD-based ensemble 4DVar. *Tellus*, **64A**, https://doi.org/10.3402/tellusa.v64i0.18375.

——, and X. B. Feng, 2015: A nonlinear least squares enhanced POD-4DVar algorithm for data assimilation. *Tellus*, **67A**, 25340, http://dx.doi.org/10.3402/tellusa.v67.25340.

——, and ——, 2017: A nonlinear least-squares-based ensemble method with a penalty strategy for computing the conditional nonlinear optimal perturbations. *Quart. J. Roy. Meteor. Soc.*, **143**, 641–649, https://doi.org/10.1002/qj.2946.

——, Z. H. Xie, and A. G. Dai, 2008: An ensemble-based explicit four-dimensional variational assimilation method. *J. Geophys. Res.*, **113**, D21124, https://doi.org/10.1029/2008JD010358.

——, ——, and Q. Sun, 2011: A POD-based ensemble four-dimensional variational assimilation method. *Tellus*, **63A**, 805–816, https://doi.org/10.1111/j.1600-0870.2011.00529.x.

——, X. B. Feng, H. Q. Zhang, B. Zhang, and R. Han, 2016: An enhanced ensemble-based method for computing CNOPs using an efficient localization implementation scheme and a two-step optimization strategy: Formulation and preliminary tests. *Quart. J. Roy. Meteor. Soc.*, **142**, 1007–1016, https://doi.org/10.1002/qj.2703.

Wang, B., J. J. Liu, S. D. Wang, W. Cheng, J. Liu, and C. S. Liu, 2010: An economical approach to four-dimensional variational data assimilation. *Adv. Atmos. Sci.*, **27**, 715–727, https://doi.org/10.1007/s00376-009-9122-3.

Whitaker, J. S., and T. M. Hamill, 2002: Ensemble data assimilation without perturbed observations. *Mon. Wea. Rev.*, **130**, 1913–1924, https://doi.org/10.1175/1520-0493(2002)130<1913:EDAWPO>2.0.CO;2.

Xu, Q., 1996: Generalized adjoint for physical processes with parameterized discontinuities. Part I: Basic issues and heuristic examples. *J. Atmos. Sci.*, **53**, 1123–1142, https://doi.org/10.1175/1520-0469(1996)053<1123:GAFPPW>2.0.CO;2.

Zhang, F. Q., M. Zhang, and J. A. Hansen, 2009: Coupling ensemble Kalman filter with four dimensional variational data assimilation. *Adv. Atmos. Sci.*, **26**, 1–8, https://doi.org/10.1007/s00376-009-0001-8.

Zhang H. Q., X. J. Tian, and C. M. Zhang, 2016: An improved localization scheme to NLS-4DVar (in Chinese). *Period. Ocean Univ. China*, **46**, 10–15, doi:10.16441/j.cnki.hdxb.20160010.

Zou, X. L., F. Vandenberghe, M. Pondeca, and Y.-H. Kuo, 1997: Introduction to adjoint techniques and the MM5 adjoint modeling system. NCAR Tech. Note NCAR/TN-435+STR, 110 pp., doi:10.5065/D6F18WNM.