ADVANCES IN ATMOSPHERIC SCIENCES, VOL. 26, NO. 1, 2009, 154–160

An Adaptive Estimation of Forecast Error Covariance Parameters for Kalman Filtering Data Assimilation

Xiaogu ZHENG*

National Institute of Water and Atmospheric Research, Wellington, New Zealand

(Received 5 December 2007; revised 22 May 2008)

ABSTRACT

An adaptive estimation of forecast error covariance matrices is proposed for Kalman filtering data assimilation. A forecast error covariance matrix is initially estimated using an ensemble of perturbation forecasts. This initially estimated matrix is then adjusted with scale parameters that are adaptively estimated by minimizing −2log-likelihood of observed-minus-forecast residuals. The proposed approach could be applied to Kalman filtering data assimilation with imperfect models when the model error statistics are not known. A simple nonlinear model (Burgers' equation model) is used to demonstrate the efficacy of the proposed approach.

Key words: data assimilation, Kalman filter, ensemble prediction, estimation

Citation: Zheng, X. G., 2009: An adaptive estimation of forecast error covariance parameters for Kalman filtering data assimilation. *Adv. Atmos. Sci.*, **26**(1), 154–160, doi: 10.1007/s00376-009-0154-5.

1. Introduction

Kalman filtering (Cohn, 1997) is a popular sequential analysis scheme for data assimilation. It is well known that estimation of forecast error covariance matrices plays a key role in the performance of the Kalman filtering schemes (e.g. Miller et al., 1994).

An approach for estimating forecast error covariance matrices in the Kalman filtering assimilation schemes is to parameterize the error covariance matrices, and then to estimate the parameters by minimizing the −2log-likelihood of observed-minus-forecast residuals (e.g., Dee and da Silva, 1999; Ozaki et al., 2000). A major obstacle for this approach is that it is difficult to parameterize forecast error covariance matrices, especially when they are not stationary in time.

A more popular approach is to generate an ensemble of perturbation forecasts by perturbing an initial state, and then the forecast error covariance matrix is estimated as the sampling covariance matrix of the ensemble (for example, Bengtsson et al., 2003). While this approach does not require parameterization of the forecast error covariance matrix, the estimation may depend on magnitude and number of the perturbations. Therefore, there is no guarantee that the sampling covariance matrix is a good estimation of the forecast error covariance matrix.

In this paper, we propose an approach for estimating forecast error covariance matrices by combining both approaches. A real-time forecast error covariance matrix is initially estimated using an ensemble of perturbation forecasts. This initially estimated matrix is then adjusted with scale parameters that are estimated by minimizing the −2log-likelihood of observed-minus-forecast residuals. The proposed scheme permits imperfect models, but knowledge of the model error covariance matrices is not required.

The paper is arranged as follows. In section 2, the details of the proposed approach are described. Sections 3 is devoted to the tests of the proposed approach using the data sets simulated by Burgers' equation model. Our conclusions are given in section 4.

2. Methodology

In this section, the proposed Kalman filtering data assimilation is outlined and a method for the adaptive estimation of the real time forecast error covariance matrices is proposed.

2.1 *Proposed Kalman filtering*

Using the notation similar to that proposed by Ide et al. (1997), a nonlinear discrete time prediction-

*Corresponding author: Xiaogu ZHENG, x.zheng@niwa.co.nz

observation system is of the form

$$\boldsymbol{x}_{i+1,\mathrm{t}} = M_i[\boldsymbol{x}_{i,\mathrm{t}}] + \boldsymbol{\eta}_i \,, \tag{1}$$

$$\boldsymbol{y}_{i,\mathrm{o}} = \boldsymbol{H}_i\boldsymbol{x}_{i,\mathrm{t}} + \boldsymbol{\varepsilon}_i \,, \tag{2}$$

where i is the time step index, the $\boldsymbol{x}_{i,\mathrm{t}}$ is the true state vector with dimension n at the time step i with t abbreviating for "true"; M_i is an prediction operator such as a numerical weather forecasting model; $\boldsymbol{y}_{i,\mathrm{o}}$ is the observation vector with dimension p_i; $\boldsymbol{H}_i$; is a $p_i \times n$ matrix indicating which linear combination of states is observed; $\boldsymbol{\eta}_i$ and $\boldsymbol{\varepsilon}_i$ are the model error and the observation error vectors, which are assumed to be statistically independent of both each other and time and have zero mean vectors and covariance matrices $\boldsymbol{Q}_i$ and $\boldsymbol{R}_i$ respectively. The goal of Kalman filtering data assimilation is to find a series of analysis states $\{\boldsymbol{x}_{i,\mathrm{a}}\}$ that is sufficiently close to the true states $\{\boldsymbol{x}_{i,\mathrm{t}}\}$ by using the information provided by the operators $\{M_i\}$ and the observations $\{\boldsymbol{y}_{i,\mathrm{o}}\}$.

Suppose an initial analysis state $\boldsymbol{x}_{0,\mathrm{a}}$ is known, our proposed Kalman filtering data assimilation comprises the following steps. Except for Step (ii) below for the estimation of the real time forecast error covariance matrices, the steps are those of the standard Kalman filter.

Step (i). Forecast the model state at time i:

$$\boldsymbol{x}_{i,\mathrm{f}} = M_{i-1}[\boldsymbol{x}_{i-1,\mathrm{a}}] \,, \tag{3}$$

where $\boldsymbol{x}_{i,\mathrm{f}}$ is assumed to be a Gaussian random vector with mean vector $\boldsymbol{x}_{i,\mathrm{t}}$ and covariance matrix $\boldsymbol{P}_{i,\mathrm{f}}$.

Step (ii). Estimate the forecast error statistics:

The forecast error covariance matrix could be estimated as the sampling covariance matrix which is defined by

$$\boldsymbol{P}_i \equiv \sum_{j=1}^{m} w_j \left({}_j\boldsymbol{x}_{i,\mathrm{f}} - \sum_{k=1}^{m} {}_k\boldsymbol{x}_{i,\mathrm{f}} w_k \right) \times \left({}_j\boldsymbol{x}_{i,\mathrm{f}} - \sum_{k=1}^{m} {}_k\boldsymbol{x}_{i,\mathrm{f}} w_k \right)^{\mathrm{T}} \tag{4}$$

where $\{\ {}_j\boldsymbol{x}_{i,\mathrm{f}}, j = 1, \dots m\}$ are the perturbation forecast states from the perturbed analysis states $\{\ {}_j\boldsymbol{x}_{i-1,\mathrm{a}}, j = 1, \dots m\}$, and $\{w_j, j = 1, \dots m\}$ are the weights. All of the existing approaches for deriving perturbed analysis states are able to be applied here, including the approach by perturbing observations used in the ensemble Kalman filtering assimilation (Anderson, 2001).

For the standard ensemble Kalman filtering assimilation, w_j is chosen as $1/m$, and $\boldsymbol{P}_i$ is regarded as an estimation of $\boldsymbol{P}_{i,\mathrm{f}}$ (Bengtsson et al., 2003). However, if $\{\ {}_j\boldsymbol{x}_{i,\mathrm{f}}, j = 1, \dots m\}$ are not samples of population of the forecast state $\boldsymbol{x}_{i,\mathrm{f}}$, and/or m is not sufficiently large, $\boldsymbol{P}_i$ can be far from the forecast error covariance matrix. To mitigate this shortfall, the estimated forecast error covariance matrix is rescaled to

$$\hat{\boldsymbol{P}}_{i,\mathrm{f}} = [\boldsymbol{\lambda}_i]\boldsymbol{P}_i[\boldsymbol{\lambda}_i] \,, \tag{5}$$

where $\boldsymbol{\lambda}_i$ is a vector representing a scale change of $\boldsymbol{P}_i$ and $[\boldsymbol{\lambda}_i]$ is a diagonal matrix with diagonal vector $\boldsymbol{\lambda}_i$. By choosing an appropriate $\boldsymbol{\lambda}_i$, $\hat{\boldsymbol{P}}_{i,\mathrm{f}}$ could be a better approximation of $\boldsymbol{P}_{i,\mathrm{f}}$ than $\boldsymbol{P}_i$ could be. The detailed estimation procedure for $\boldsymbol{\lambda}_i$ is proposed in this paper, and is documented in section 2.2. Intuitively, $[\boldsymbol{\lambda}_i]$ can be viewed as a multivariate covariance inflation operator. While the common covariance inflation factor is a scalar value and is estimated by trials (Constantinescu et al., 2007), this study extends it to the diagonal version, and also optimizes it.

Step (iii). Calculate the observed-minus-forecast residuals:

$$\boldsymbol{d}_i = \boldsymbol{y}_{i,\mathrm{o}} - \boldsymbol{H}_i\boldsymbol{x}_{i,\mathrm{f}} \,, \tag{6}$$

where $\boldsymbol{d}_i$ is assumed to be Gaussian with a zero mean vector and covariance matrix $\boldsymbol{H}_i[\boldsymbol{\lambda}_i]\boldsymbol{P}_i[\boldsymbol{\lambda}_i]\boldsymbol{H}_i^{\mathrm{T}} + \boldsymbol{R}_i$.

Step (iv). Calculate the analysis state:

$$\boldsymbol{x}_{i,\mathrm{a}} = \boldsymbol{x}_{i,\mathrm{f}} + \hat{\boldsymbol{P}}_{i,\mathrm{f}}\boldsymbol{H}_i^{\mathrm{T}}(\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}}\boldsymbol{H}_i^{\mathrm{T}} + \boldsymbol{R}_i)^{-1}\boldsymbol{d}_i \tag{7}$$

Step (v). If $\boldsymbol{y}_{i,0}$ is not the last observation, put $i = i + 1$ and return to step (i). Otherwise, stop the filtering. $\boldsymbol{x}_{i,\mathrm{a}}$ are the filtered states. Each $\boldsymbol{x}_{i,\mathrm{a}}$ is Gaussian with mean $\boldsymbol{x}_{i,\mathrm{t}}$ and covariance matrices

$$\boldsymbol{P}_{i,\mathrm{a}} = \hat{\boldsymbol{P}}_{i,\mathrm{f}} - \hat{\boldsymbol{P}}_{i,\mathrm{f}}\boldsymbol{H}_i^{\mathrm{T}}(\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}}\boldsymbol{H}_i^{\mathrm{T}} + \boldsymbol{R}_i)^{-1}\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}} \,. \tag{8}$$

From Eq. (7), $\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}}$ and $\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}}\boldsymbol{H}_i^{\mathrm{T}}$ are sufficient to determine $\boldsymbol{x}_{i,\mathrm{a}}$. They can be estimated by

$$\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}} = \sum_{j=1}^{m} w_j \left(\boldsymbol{H}_i[\boldsymbol{\lambda}_i]({}_j\boldsymbol{x}_{i,\mathrm{f}} - \sum_{k=1}^{m} {}_k\boldsymbol{x}_{i,\mathrm{f}} w_k) \right) \times \left([\boldsymbol{\lambda}_i]({}_j\boldsymbol{x}_{i,\mathrm{f}} - \sum_{k=1}^{m} {}_k\boldsymbol{x}_{i,\mathrm{f}} w_k) \right)^{\mathrm{T}} \tag{9}$$

and

$$\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}}\boldsymbol{H}_i^{\mathrm{T}} = \sum_{j=1}^{m} w_j \left(\boldsymbol{H}_i[\boldsymbol{\lambda}_i]({}_j\boldsymbol{x}_{i,\mathrm{f}} - \sum_{k=1}^{m} {}_k\boldsymbol{x}_{i,\mathrm{f}} w_k) \right) \times \left(\boldsymbol{H}_i[\boldsymbol{\lambda}_i]({}_j\boldsymbol{x}_{i,\mathrm{f}} - \sum_{k=1}^{m} {}_k\boldsymbol{x}_{i,\mathrm{f}} w_k) \right)^{\mathrm{T}} . \tag{10}$$

Usually, the observational dimension p_i is significantly less than the model dimension n. Then the computational cost for $\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}}$ and $\boldsymbol{H}_i\hat{\boldsymbol{P}}_{i,\mathrm{f}}\boldsymbol{H}_i^{\mathrm{T}}$ is more economical than that for $\hat{\boldsymbol{P}}_{i,\mathrm{f}}$.

2.2 *Estimation of scale parameter* $\boldsymbol{\lambda}_i$

We estimate $\boldsymbol{\lambda}_i$ by minimizing the -2log-likelihood of the observed-minus-forecast residual $\boldsymbol{d}_i = \boldsymbol{y}_{i,\mathrm{o}} - \boldsymbol{H}_i\boldsymbol{x}_{i,\mathrm{f}}$. Since $\boldsymbol{d}_i$ is assumed to be Gaussian with a zero mean vector and covariance matrix $\boldsymbol{H}_i[\boldsymbol{\lambda}_i]\boldsymbol{P}_i[\boldsymbol{\lambda}_i]\boldsymbol{H}_i^{\mathrm{T}} + \boldsymbol{R}_i$ [Eq. (2)], its -2log-likelihood function is

$$-2L_i(\boldsymbol{\lambda}_i) = \ln[\det(\boldsymbol{H}_i[\boldsymbol{\lambda}_i]\boldsymbol{P}_i[\boldsymbol{\lambda}_i]\boldsymbol{H}_i^{\mathrm{T}} + \boldsymbol{R}_i)] + \boldsymbol{d}_i^{\mathrm{T}}(\boldsymbol{H}_i[\boldsymbol{\lambda}_i]\boldsymbol{P}_i[\boldsymbol{\lambda}_i]\boldsymbol{H}_i^{\mathrm{T}} + \boldsymbol{R}_i)^{-1}\boldsymbol{d}_i \ , \quad (11)$$

where det represents the determinant of a matrix (Dee and da Silva, 1999; Ozaki et al., 2000).

Vector $\boldsymbol{\lambda}_i$ is comprised of the two components: the observable component $\boldsymbol{\lambda}_{i,\mathrm{o}}$ (i.e., $\boldsymbol{H}_i[\boldsymbol{\lambda}_i]$ depends on $\boldsymbol{\lambda}_{i,\mathrm{o}}$), and an unobservable component $\boldsymbol{\lambda}_{i,\mathrm{u}}$ (i.e., $\boldsymbol{H}_i[\boldsymbol{\lambda}_i]$ is independent of $\boldsymbol{\lambda}_{i,\mathrm{u}}$). From Eq. (11), the -2log-likelihood function is the function of $\boldsymbol{H}_i[\boldsymbol{\lambda}_i]$. Then it is only the function of the observable component $\boldsymbol{\lambda}_{i,\mathrm{o}}$. Therefore, the unobservable component $\boldsymbol{\lambda}_{i,\mathrm{u}}$ cannot be estimated by minimizing function (11). These two components should be estimated separately.

2.2.1 *Estimate observable component* $\boldsymbol{\lambda}_{i,\mathrm{o}}$

The observable component $\boldsymbol{\lambda}_{i,\mathrm{o}}$ can be estimated by minimizing $-2L_i(\boldsymbol{\lambda}_i)$ ([see Eq. (11)]. To do this effectively, we need to calculate its first derivative $\nabla_\lambda L_i(\boldsymbol{\lambda})$ and its second derivative $\nabla^2_{\boldsymbol{\lambda}} L_i(\boldsymbol{\lambda})$. Then, the fastest descendent direction of the likelihood function is

$$\delta\boldsymbol{\lambda} = -\nabla_{\boldsymbol{\lambda}} L_i(\boldsymbol{\lambda})[\nabla^2_{\boldsymbol{\lambda}} L_i(\boldsymbol{\lambda})]^{-1} \ . \quad (12)$$

For this purpose, we introduce the following matrix notation. For a matrix $\boldsymbol{A}$, the i-th column is denoted as $\boldsymbol{a}_i$ and its (i,j)-th entry is denoted as a_{ij}. Suppose matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ have a same dimension. $\boldsymbol{A}\times\boldsymbol{B}$ represents the matrix with (i,j)-th entry $a_{ij}b_{ij}$ (i.e., element-by-element or Shur product).

Under these notations, the first derivative of $L_i(\boldsymbol{\lambda})$ at time step i is

$$\nabla_\lambda[-L_i(\boldsymbol{\lambda})] = \begin{pmatrix} (\partial/\partial\lambda_1)(-L_i(\boldsymbol{\lambda})) \\ (\partial/\partial\lambda_2)(-L_i(\boldsymbol{\lambda})) \\ \cdots \\ (\partial/\partial\lambda_n)(-L_i(\boldsymbol{\lambda})) \end{pmatrix} = -\begin{pmatrix} (\boldsymbol{p}_1^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_1 \\ (\boldsymbol{p}_2^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_2 \\ \cdots \\ (\boldsymbol{p}_n^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_n \end{pmatrix} + \boldsymbol{c}\times\begin{pmatrix} (\boldsymbol{p}_1^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} \\ (\boldsymbol{p}_2^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} \\ \cdots \\ (\boldsymbol{p}_n^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} \end{pmatrix} , \quad (13)$$

where $\boldsymbol{P} \equiv \boldsymbol{P}_i, \boldsymbol{Q} \equiv \boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{H}$, and $\boldsymbol{c} \equiv \boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{d}$ with $\boldsymbol{H} = \boldsymbol{H}_i, \boldsymbol{d} = \boldsymbol{d}_i$ and $\boldsymbol{R} = \boldsymbol{R}_i$. The detailed proof is documented in the Appendix.

The second derivative of $L_i(\boldsymbol{\lambda})$ [the Hessian matrix $\nabla^2_\lambda L_i(\boldsymbol{\lambda})$] is

$$\begin{aligned} &\frac{\partial}{\partial\boldsymbol{\lambda}}\begin{pmatrix} \boldsymbol{p}_1^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_1 & \boldsymbol{p}_2^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_2 & \cdots & \boldsymbol{p}_n^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_n \end{pmatrix} \\ &- \frac{\partial}{\partial\boldsymbol{\lambda}}\begin{pmatrix} (\boldsymbol{c}^{\mathrm{T}}\times(\boldsymbol{p}_1^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} & \boldsymbol{p}_2^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} & \cdots & \boldsymbol{p}_n^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c}) \end{pmatrix} \\ &= \boldsymbol{Q}\times\boldsymbol{P} - (\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q})\times(\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P}) - (\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q})[\boldsymbol{\lambda}]\boldsymbol{P})\times\boldsymbol{Q} \\ &+ \boldsymbol{Q}\times(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{P}) + (\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P})\times(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}) - \\ &\boldsymbol{P}\times(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}) + (\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q})\times(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{P}) + \\ &(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P})\times(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}) \end{aligned} \quad (14)$$

The detailed proof is also documented in the Appendix. Note that both derivatives also only depend on $\boldsymbol{\lambda}_{i,\mathrm{o}}$.

After the fastest descendent direction is obtained, $\boldsymbol{\lambda}_i$ is substituted by $\boldsymbol{\lambda} + \mu\delta\boldsymbol{\lambda}_i$, where $0 < \mu \leqslant 1$ is a scalar to guarantee that $-2L_i(\boldsymbol{\lambda}_i + \mu\delta\boldsymbol{\lambda}_i)$ is less than $-2L_i(\boldsymbol{\lambda})$.

2.2.2 *Estimate unobservable component* $\boldsymbol{\lambda}_{i,\mathrm{u}}$

Although the likelihood at the time step i is independent of the unobservable component $\boldsymbol{\lambda}_{i,\mathrm{u}}$, the -2log-likelihood at time step $i+1(-2L_{i+1})$ does depend on $\boldsymbol{\lambda}_{i,\mathrm{u}}$. This is because L_{i+1} depends on $\boldsymbol{x}_{i,\mathrm{a}}$, the analysis state at time step i. From Eq. (7),

$$\boldsymbol{x}_{i,\mathrm{a}} = \boldsymbol{x}_{i,\mathrm{f}} + [\boldsymbol{\lambda}_i]\boldsymbol{P}_i[\boldsymbol{\lambda}_i]\boldsymbol{H}_i^{\mathrm{T}}(\boldsymbol{H}_i[\boldsymbol{\lambda}_i]\boldsymbol{P}_i[\boldsymbol{\lambda}_i]\boldsymbol{H}_i^{\mathrm{T}} + \boldsymbol{R}_i)^{-1}\boldsymbol{d}_i \ , \quad (15)$$

which depends on both $\boldsymbol{\lambda}_{i,\mathrm{o}}$ and $\boldsymbol{\lambda}_{i,\mathrm{u}}$.

We shall estimate $\boldsymbol{\lambda}_{i,\mathrm{u}}$ and $\boldsymbol{\lambda}_{i+1,\mathrm{o}}$ jointly. For a $\boldsymbol{\lambda}_{i+1,\mathrm{o}}$, find a new $\boldsymbol{\lambda}_{i,\mathrm{u}}$ to reduce $-2L_{i+1}$ by direct search. Then, apply the procedure documented in section 2.2.1 to find a new $\boldsymbol{\lambda}_{i+1,\mathrm{o}}$ that further reduces $-2L_{i+1}$. Continue this procedure iteratively until L_{i+1} converges.

2.2.3 *Constraint on parameter* $\boldsymbol{\lambda}$

The dimension of $\boldsymbol{\lambda}$ is equivalent to the dimension of analysis states (n), that is often too large to estimate $\boldsymbol{\lambda}$. In practice, constraints are often imposed on $\boldsymbol{\lambda}$. As an example, we may assume components of $\boldsymbol{\lambda}$ at one vertical level are all the same in meteorological data assimilation.

The first derivative and the second derivative of the constrained $L(\boldsymbol{\lambda})$ can be easily derived from the first derivative and second derivative of

the unconstrained $L(\boldsymbol{\lambda})$. Suppose the analysis space is partitioned into $N(N \leqslant n)$ blocks, i.e., $\boldsymbol{\lambda}_N = (\lambda_1, \cdots, \lambda_1, \lambda_2, \cdots, \lambda_2, \cdots, \lambda_N, \cdots, \lambda_N)$ Then the first derivative of constrained $L(\boldsymbol{\lambda}_N)$ is the N dimensional vector which element for a block is the sum of elements of $\nabla_\lambda L(\boldsymbol{\lambda}_N)$ within that block. Similarly, the second derivative of constrained $L(\boldsymbol{\lambda}_N)$ is the N by N matrix which element for a block pair is the sums of element of $\nabla^2_\lambda L(\boldsymbol{\lambda}_N)$ within the block pairs.

Because the constraint has always to be applied for large models, the dimension of the second derivatives $\nabla^2_\lambda L(\boldsymbol{\lambda})$ will not be large. Therefore, the inverse of $\nabla^2_\lambda L(\boldsymbol{\lambda})$ in Eq. (12) can be calculated directly.

2.3 *Computing* $(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1}$

$(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}$ can be calculated as follows. Decompose $\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}$ into

$$\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}} = \sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}} , \tag{16}$$

where

$$\boldsymbol{z}_j = \boldsymbol{H}[\boldsymbol{\lambda}]\sqrt{w_j}\left({}_j\boldsymbol{x}_{i,f} - \sum_{k=1}^{m} {}_k\boldsymbol{x}_{i,f} w_k\right) . \tag{17}$$

Define $\boldsymbol{A}_i = \sum_{k=1}^{i} \boldsymbol{z}_k \boldsymbol{z}_k^{\mathrm{T}} + \boldsymbol{R}$. It is easy to check that

$$\boldsymbol{A}_1^{-1} = \boldsymbol{R}^{-1} - (\boldsymbol{R}^{-1}\boldsymbol{z}_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{R}^{-1})/(1+\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{z}_1) , \tag{18}$$

and for $1 \leqslant j \leqslant m-1$

$$\boldsymbol{A}_{j+1}^{-1} = \boldsymbol{A}_j^{-1} - (\boldsymbol{A}_j^{-1}\boldsymbol{z}_j\boldsymbol{z}_j^{\mathrm{T}}\boldsymbol{A}_j^{-1})/(1+\boldsymbol{z}_j^{\mathrm{T}}\boldsymbol{A}_j^{-1}\boldsymbol{z}_j) . \tag{19}$$

Thus, $(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1} = \boldsymbol{A}_m^{-1}$ can be calculated iteratively.

In this way, the inverse $(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1}$ can be calculated without significantly computational cost, providing the inverse of the observational matrix $\boldsymbol{R}^{-1}$ is known. However, $\boldsymbol{R}^{-1}$ is also required for any variational approach. Most ensemble based Kalman filtering are likely to be as computationally expensive as 4D-VAR, and perhaps significantly more expensive when there are an overwhelmingly large number of observations, such as very high resolution satellite images (Hamill, 2006). Our proposed approach for calculating the inverse provides a solution to overcome this obstacle.

3. Application to Burgers' equation model

3.1 *Burgers' equation model*

Burgers (1974) proposed the following equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2} \tag{20}$$

to describe the one-dimensional advection-diffusion process over an infinite spatial domain. Recently, Zhu and Kamachi (2000) used it as a test bed for several data assimilation schemes. In the present study, we also use it to test our proposed methodology. For the readers' convenience, we briefly introduce how to simulate the solution of a Burgers equation following Zhu and Kamachi (2000). This solution is then used as the series of true states in the present study. Knowing the true states, the root mean square error (RMSE) of the assimilated analysis states can be estimated.

The numerical model is defined as a finite-difference leapfrog scheme with a forward step every 15 time steps. This simple scheme gives accurate results by comparing with one analytical solution (Uboldi and Kamachi, 2000). We will use the following abbreviations: m for meter, km for kilometer, s for second and h for hour. The computation spatial domain is [−1000, 1000] km, but only the solution within [−100, 150] km is considered. A large computation domain is used to reduce the boundary effects on the interior solution. The model parameters are: time step $\Delta t = 60$ s; spatial grid resolution $\Delta x = 5$ km. Therefore, n (the dimension of state vector) is $51(= 150/5 + 1 + 100/5)$. The simulation time is $T = 16$ h. The "true" solution is generated by running the model with the initial condition

$$u(x,0) = \begin{cases} 0 & x < -L;\ \ x > L \\ u_0(1+2x/L)(1-2x/L)^2 & 0 \leqslant x < L \\ u_0(1-2x/L)(1+2x/L)^2 & -L < x < 0 \end{cases} \tag{21}$$

where $u_0 = 5$ m s^{-1}, and $L = 50$ km and with diffusion coefficient $v = 1.0 \times 10^4$ m^2 s^{-1}. Denote the true solution by $u_{\mathrm{t}}(x,t)$, where -100 km$\leqslant x \leqslant 150$ km and 0 h$\leqslant t \leqslant 16$ h.

In the data assimilation experiments with the imperfect model, the diffusion coefficient is set to be $v = 1.4 \times 10^4$ m^2 s^{-1} to simulate the model error. The model error caused by the wrong diffusion coefficient is estimated by running the model twice with the same above initial condition, but with different diffusion coefficients. The spatially averaged root mean square of the model error is shown in Fig. 2 of Zhu and Kamachi (2000) with an overall averaged error of about 0.13 m s^{-1}. The overall averaged signal is about 0.9 m s^{-1} in the time-space domain. The averaged model error is about 14% of the average signal. The observations are assumed to be available at every other model grid point from $x = -100$ km to 150 km. In total there are 26 observation stations (total of 51 model grid points). The observations are available only at hour 1, 4, 7 and

10, and are statistically independent of each other with standard deviation 0.15 m s^{-1}.

3.2 *Results*

To construct $\boldsymbol{P}_i$ in Eq. (4), the number of perturbations m and the weights $\{w_j, j = 1, \cdots, m\}$ are selected similar to that in the unscented ensemble Kalman filter (Julier and Uhlmann, 2004), where $m = 2n + 1$ and

$$w_j = \begin{cases} k/(n+k) & j = 0\,, \\ 1/2(n+k) & j = 2, \cdots, n+1\,, \end{cases} \tag{22}$$

where k is a parameter. Julier and Uhlmann (2004) suggested that if the forecast error is Gaussian, set $n + k = 3$. Otherwise set $n + k < 3$. Since the forecast error is assumed Gaussian and $n = 51$, we set $k = -48$. The perturbed analysis states is selected as

$${}_j\boldsymbol{x}_{i,\mathrm{a}} = \begin{cases} \boldsymbol{x}_{i,\mathrm{a}} & j = 1\,, \\ \boldsymbol{x}_{i,\mathrm{a}} + \alpha\boldsymbol{\delta}_j & j = 2, \cdots, n+1\,, \\ \boldsymbol{x}_{i,\mathrm{a}} - \alpha\boldsymbol{\delta}_j & j = n+2, \cdots, 2n+1\,, \end{cases} \tag{23}$$

where $\boldsymbol{\delta}_j$ is j-th column of the n by n identity matrix and $\alpha > 0$ is the perturbation parameter. Here α is set as 0.1 m s^{-1} to match the overall averaged initial error of 0.13 m s^{-1} (Zhu and Kamachi, 2000). Scale parameters for the 26 observational points $\boldsymbol{\lambda}_{i,\mathrm{o}}$ and the scale parameters for the other 25 unobservable points $\boldsymbol{\lambda}_{i,\mathrm{u}}$ are constraint as one parameter respectively.

The estimated scale parameters, the -2log-likelihood, and the RMSE of the analysis state at each time step with observation are listed in Table 1. The RMSE of the analysis state from hour 1 to hour 16 are plotted in Fig. 1. As a comparison, the -2log-likelihood, and the RMSE of analysis without the parameter adjustment (i.e., $\boldsymbol{\lambda} = \boldsymbol{I}$) are listed in Table 2, and the corresponding RMSE are also plotted in Fig. 1.

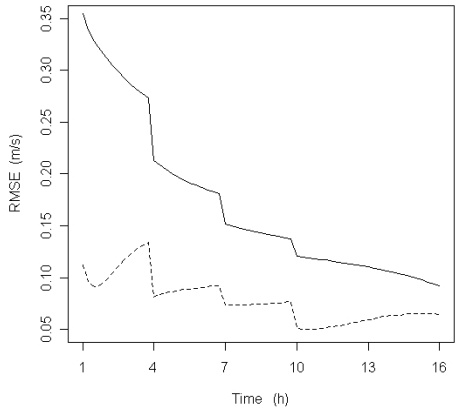

Fig. 1. Root mean square errors of the analysis states with the parameter adjustment (dashed) and without the parameter adjustment (solid).

Table 1. Results with the parameter adjustment.

	Hour 1	Hour 4	Hour 7	Hour 10
RMSR (m s^{-1})	0.12	0.097	0.068	0.04
$-2\log\,[L_i(\lambda_i)]$	-389	-421	-429	-436
Observable $\lambda_{i,\mathrm{o}}$	10	2	4	0.5
Unobservable $\lambda_{i,\mathrm{o}}$	11	1	4	$*$

Note: Unobserved $\lambda_{i,\mathrm{o}}$ is not required at the last time step.

Table 2. Results without the parameter adjustment.

	Hour 1	Hour 4	Hour 7	Hour 10
RMSR (m s^{-1})	0.35	0.19	0.14	0.09
$-2\log\,[L_i(\boldsymbol{I})]$	-251	-387	-411	-421

Tables 1 and 2 show that the -2log-likelihood of the observed-minus-forecast residual with the parameter adjustment is significantly less than those without parameter adjustment. Correspondingly, the RMSE for the analysis states with the parameter adjustment are less than the half of those without the parameter adjustment. This fact is also shown in Fig. 1.

Zhu and Kamachi (2000) proposed a number of adaptive variational assimilation schemes with imperfect models. They tested their schemes against the simplified 4D-VAR assimilation scheme and the space variable optimal nudging assimilation scheme, using Burgers' equation model as the test bed. The RSME derived by these schemes are shown in Fig. 3a of Zhu and Kamachi (2000). For these schemes, the RMSE for the reduced order adaptive variational method (ROAV) is the best one. Comparing the RMSE derived by ROAV with the RMSE shown in Fig. 1 and in Table 1, our RMSE is comparable to theirs at early hours (1 and 4), but ours is 0.03 m s^{-1} less (at hour 7) and 0.01 m s^{-1} less (at hour 10) than theirs. Furthermore there are only seven parameters (Table 1) to be estimated in our proposed schemes, while there are 51 parameters in ROAV. These facts indicate that the adaptive estimation of forecast error statistic proposed in this paper is at least a competitive scheme.

4. Conclusions

An adaptive estimation of forecast error statistics is proposed for Kalman filtering data assimilation for non-linear imperfect models. It has the advantage that the model error statistics do not need to be known and the forecast error statistics and observational error statistics can depend on time steps. The proposed scheme may have potential in data assimilation with large models. It is showed, by a case study, that the proposed inflation can improve the assimilation. In the future, we plan to further study the possibility of appling the proposed inflation approach to improve the

assimilation when the number of ensemble members is small.

Acknowledgements. This study was initiated during my visit to the Institute of Statistical Mathematics (ISM) in Tokyo under the support of a Visiting Fellowship offered by the Japanese Ministry of Education, Science, Sports and Culture. The study has been continued under the support of the Foundation for Research Science and Technology of New Zealand under contract C01X0401. I thank Prof. K. Tanabe for his collaboration on this paper and Prof. T. Ozaki for his support and many stimulating discussions. We would like to thank Prof. Zhidong Bai and Dr. Shouzheng Tang for their comments on mathematical algorithms, Prof. Masahide Kimoto and Dr. Phil Andrews for their comments on physical sciences and the two anonymous reviewers for improving the presentation.

APPENDIX

Proof of the Derivatives of the Log-Likelihood

1. Proof of the first derivative of the log-likeli hood

Let subscripts r and s represent components of the model state vector.

$$\frac{\partial}{\partial\lambda_r}\{\ln[\det(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P})[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})]\}$$

$$=\mathrm{tr}\left[(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\frac{\partial}{\partial\lambda_r}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})\right]$$

$$=\mathrm{tr}\left[(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{H}([\boldsymbol{\delta}_r]\boldsymbol{P}[\boldsymbol{\lambda}]+[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_r])\boldsymbol{H}^{\mathrm{T}}\right]$$

$$=\mathrm{tr}\left[(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{H}[\boldsymbol{\delta}_r]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}\right]$$

$$+\mathrm{tr}\left[(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_r]\boldsymbol{H}^{\mathrm{T}}\right]$$

$$=2\mathrm{tr}\left[(\boldsymbol{P}[\boldsymbol{\lambda}][\boldsymbol{\delta}_r](\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{H})\right]$$

$$=2\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_r\,. \tag{A1}$$

$$\frac{\partial}{\partial\lambda_r}[\boldsymbol{d}^{\mathrm{T}}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{d}]$$

$$=-\boldsymbol{d}^{\mathrm{T}}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\times$$

$$\frac{\partial}{\partial\lambda_r}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{d}$$

$$=-[\boldsymbol{d}^{\mathrm{T}}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{H})([\delta_r]\boldsymbol{P}[\boldsymbol{\lambda}]+$$

$$+[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_r])(\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{d})$$

$$=-\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{\delta}_r]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{c}-\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_r]\boldsymbol{c}$$

$$=-c_r\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c}-\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{p}_r c_r$$

$$=-2c_r\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c}\,. \tag{A2}$$

From Eq. (11) and Eqs. (A1)–(A2)

$$\frac{\partial}{\partial\lambda_r}L_i(\boldsymbol{\lambda})=-\frac{1}{2}(2\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_r-2c_r\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c})$$

$$=-\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_r+c_r\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c}\,.$$

Thus, the first derivative [i.e., Eq. (13)] is derived.

2. Proof of the second derivative of the log-likelihood

$$\frac{\partial}{\partial\lambda_s}\boldsymbol{Q}=\boldsymbol{H}^{\mathrm{T}}\left\{\frac{\partial}{\partial\lambda_s}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\right\}\boldsymbol{H}$$

$$=-\boldsymbol{H}^{\mathrm{T}}\{(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{H}([\boldsymbol{\delta}_s]\boldsymbol{P}[\boldsymbol{\lambda}]+$$

$$[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_s])\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}([\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\}\boldsymbol{H}$$

$$=-\boldsymbol{Q}([\boldsymbol{\delta}_s]\boldsymbol{P}[\boldsymbol{\lambda}]+[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_s])\boldsymbol{Q}\,.$$

Then

$$\frac{\partial}{\partial\lambda_s}\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_r$$

$$=\boldsymbol{p}_r^{\mathrm{T}}\left(\frac{\partial}{\partial\lambda_s}[\boldsymbol{\lambda}]\right)\boldsymbol{q}_r+\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\frac{\partial}{\partial\lambda_s}\boldsymbol{q}_r$$

$$=\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\delta}_s]\boldsymbol{q}_r-\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{Q}[\boldsymbol{\delta}_s]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{q}_r-$$

$$\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_s]\boldsymbol{q}_r$$

$$=p_{rs}q_{sr}-(\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_s)(\boldsymbol{p}_s^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_r)-$$

$$(\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{p}_s)q_{sr}\,.$$

Thus,

$$\frac{\partial}{\partial\boldsymbol{\lambda}}\left(\begin{array}{cccc}\boldsymbol{p}_1^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_1 & \boldsymbol{p}_2^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_2 & \cdots & \boldsymbol{p}_n^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{q}_n\end{array}\right)$$

$$=\boldsymbol{Q}\times\boldsymbol{P}-(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q}\times(\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P})-$$

$$(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P})\times\boldsymbol{Q} \tag{A3}$$

Since

$$\frac{\partial}{\partial\lambda_s}\boldsymbol{c} = \boldsymbol{H}^{\mathrm{T}}\left\{\frac{\partial}{\partial\lambda_s}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\right\}\boldsymbol{d}$$

$$= -\boldsymbol{H}^{\mathrm{T}}\{(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{H}([\boldsymbol{\delta}_s]\boldsymbol{P}[\boldsymbol{\lambda}]+$$

$$[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_s])\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\}\boldsymbol{d}$$

$$= -\boldsymbol{Q}([\boldsymbol{\delta}_s]\boldsymbol{P}[\boldsymbol{\lambda}]+[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_s])\boldsymbol{c}\,,$$

it follows that

$$\frac{\partial}{\partial\lambda_s}c_r\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c}$$

$$= \left(\frac{\partial}{\partial\lambda_s}c_r\right)\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} + c_r\boldsymbol{p}_r^{\mathrm{T}}\left(\frac{\partial}{\partial\lambda_s}[\boldsymbol{\lambda}]\right)\boldsymbol{c}+$$

$$c_r\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\left(\frac{\partial}{\partial\lambda_s}\boldsymbol{c}\right)$$

$$= -\boldsymbol{q}_r^{\mathrm{T}}([\boldsymbol{\delta}_s]\boldsymbol{P}[\boldsymbol{\lambda}]+[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_s])\boldsymbol{c}\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} + c_r\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\delta}_s]\boldsymbol{c}$$

$$- c_r\boldsymbol{p}_r^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{Q}([\boldsymbol{\delta}_s]\boldsymbol{P}[\boldsymbol{\lambda}]+[\boldsymbol{\lambda}]\boldsymbol{P}[\boldsymbol{\delta}_s])\boldsymbol{c}$$

$$= -q_{rs}(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{P})_{sr} - (\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P})_{rs}(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}})_{rs}+$$

$$\boldsymbol{p}_{rs}(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}})_{rs} - (\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q})_{rs}(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}})_{sr}-$$

$$(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P})_{rs}(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}})_{sr}$$

Therefore

$$\frac{\partial}{\partial\boldsymbol{\lambda}}\left(\begin{array}{cccc}\boldsymbol{c}^{\mathrm{T}}\times(\boldsymbol{p}_1^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} & \boldsymbol{p}_2^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c} & \cdots & \boldsymbol{p}_n^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{c})\end{array}\right)$$

$$= -\boldsymbol{Q}\times(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{P})-$$

$$(\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P})\times(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}})+$$

$$\boldsymbol{P}\times(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}) - (\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q})\times(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{\lambda}]\boldsymbol{P})-$$

$$(\boldsymbol{P}[\boldsymbol{\lambda}]\boldsymbol{Q}[\boldsymbol{\lambda}]\boldsymbol{P})\times(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}) \qquad \text{(A4)}$$

The Hessian matrix (Eq. 14) is derived from Eqs. (A3)–(A4).

REFERENCES

Anderson, J. L., 2001: An ensemble adjustment Kalman filter for data assimilation. *Mon. Wea., Rev.*, **129**, 2844–2903.

Bengtsson, T., D. Nychka, and C. Snyder, 2003: A frame work for data assimilation and forecasting in high dimensional non-linear dynamic systems. *J. Geophys. Res.*, **108**(D), 8875.

Burgers, J. M., 1974: *The Nonlinear Diffusion*. D. Reidel Publ. Co., Dordrecht, Holland, 173pp.

Cohn, S. E. 1997: An introduction to estimation theory. *J. Meteor. Soc. Japan*, **75**, 257–288.

Constantinescu, M., A. Sandu, T. Chai, and G. R. Carmichael, 2007: Ensemble-based chemical data assimilation. I: General approach. *Quart. J. Roy. Meteor. Soc.*, **133**, 1229–1243.

Dee, D. P., and A. M. da Silva, 1999: Maximum-likelihood estimation of forecast and observation error covariance parameters. Part 1: Methodology. *Mon. Wea. Rev.*, **127**, 1822–1849.

Hamill, T. M., 2006: Ensemble-based atmospheric data assimilation. *Predictability of Weather and Climate*, Cambridge press, 123–156.

Ide, K., P. Courtier, G. Michael, and A. C. Lorenc, 1997: Unified notation for data assimilation: operational, sequential and variational. *J. Meteor. Soc. Japan*, **75**, 71–79.

Julier, S. J., and K. Uhlmann, 2004: Unscented filtering and nonlinear estimation. *Proc. IEEE Aeroscience and Electronic Systems*, **92**, 410–422.

Miller, R. N., M. Ghil, and F. Gauthiez, 1994: An advanced data assimilation in strongly nonlinear dynamical systems. *J. Atmos. Sci.*, **15**, 1037–1056.

Ozaki, T., J. C. Jimenez, and V. H. Ozaki, 2000: The role of the likelihood function in the estimation of chaos models. *Journal of Time Series Analysis*, **21**, 363–387.

Uboldi, F., and M. Kamachi, 2000: Time-space weak-constraint data assimilation for nonlinear models. *Tellus*, **52A**, 412–421.

Zhu, J. and M. Kamachi, 2000: An adaptive variational method for data assimilation with imperfect models. *Tellus*, **52A**, 265–279.