

## Variance Reduced Ensemble Kalman Filtering

A. W. HEEMINK, M. VERLAAN, AND A. J. SEGERS

*Faculty of Information Technology and Systems, Department of Applied Mathematical Analysis, Delft University of Technology, Delft, Netherlands*

(Manuscript received 3 February 2000, in final form 14 November 2000)

### ABSTRACT

A number of algorithms to solve large-scale Kalman filtering problems have been introduced recently. The ensemble Kalman filter represents the probability density of the state estimate by a finite number of randomly generated system states. Another algorithm uses a singular value decomposition to select the leading eigenvectors of the covariance matrix of the state estimate and to approximate the full covariance matrix by a reduced-rank matrix. Both algorithms, however, still require a huge amount of computer resources. In this paper the authors propose to combine the two algorithms and to use a reduced-rank approximation of the covariance matrix as a variance reducer for the ensemble Kalman filter. If the leading eigenvectors explain most of the variance, which is the case for most applications, the computational burden to solve the filtering problem can be reduced significantly (up to an order of magnitude).

### 1. Introduction

Kalman filtering is a powerful framework for solving data assimilation problems (see Ghil and Malanotte-Rizzoli 1991). In order to use a Kalman filter for assimilating data into a numerical model, this model is embedded into a stochastic environment by introducing a system noise process. In this way it is possible to take into account the inaccuracies of the underlying deterministic system. By using a Kalman filter, the information provided by the resulting stochastic–dynamic model and the noisy measurements are combined to obtain an optimal estimate of the state of the system. The standard Kalman filter implementation, however, would impose an unacceptable computational burden. In order to obtain a computationally efficient filter, simplifications have to be introduced.

The ensemble Kalman filter (EnKF) was introduced by Evensen (1994) and has been used successfully in many applications (see Evensen and Van Leeuwen 1996; Houtekamer and Mitchel 1998; Cañizares 1999). This Monte Carlo approach is based on a representation of the probability density of the state estimate by a finite number  $N$  of randomly generated system states. The algorithm does not require a tangent linear model and is very easy to implement. The computational effort required for the EnKF is approximately  $N$  times as much as the effort required for the underlying model. The only

serious disadvantage is that the statistical error in the estimates of the mean and covariance matrix from a sample decreases very slowly for larger sample size. This is a well-known fundamental problem with all Monte Carlo methods. As a result, for most practical problems the sample size chosen has to be rather large. Here it should be noted that a properly constructed ensemble Kalman filter can still provide an improved analysis even with small-sized ensembles (see Houtekamer and Mitchel 1998).

Another approach to solve large-scale Kalman filtering problems is to approximate the full covariance matrix of the state estimate by a matrix with reduced rank. This approach was introduced by Cohn and Todling (1995, 1996) and Verlaan and Heemink (1995, 1997), where the latter used a robust square root formulation for the filter implementation. Algorithms based on similar ideas have been proposed and applied by Lermusiaux (1997) and Pham et al. (1998).

The reduced-rank approaches can also be formulated as an ensemble Kalman filter where the  $q$  ensemble members have not been chosen randomly, but in the directions of the  $q$  leading eigenvectors of the covariance matrix (see Verlaan and Heemink 1997). As a result also these algorithms do not require a tangent linear model. The computational effort required is approximately  $q + 1$  model simulations plus the computations required for the singular value decomposition to determine the leading eigenvectors [ $O(q^3)$ ; see Heemink et al. (1997)]. In many practical problems the full covariance can be approximated accurately by a reduced-rank matrix with relatively small value of  $q$ . However, re-

---

*Corresponding author address:* A. W. Heemink, Delft University of Technology, Department of Applied Mathematical Analysis, Mekelweg 4, 2628 CD Delft, Netherlands.  
E-mail: a.w.heemink@math.tudelft.nl

duced-rank approaches often suffer from filter divergence problems for small values of  $q$ . This was observed first by Cohn and Todling (1995), who tried several methods of compensating for the truncation error. The main reason for the occurrence of filter divergence is the fact that truncation of the eigenvectors of the covariance matrix implies that the covariance is always underestimated. It is well known that underestimating the covariance may cause filter divergence. Filter divergence can be avoided by choosing  $q$  relatively large, but this of course reduces the computational efficiency of the method considerably.

Cañizares (1999) compared the EnKF and the reduced-rank approach for a number of practical large-scale shallow water flow problems. His conclusion is that the computational efficiency of both approaches is comparable. The rank  $q$  of the reduced-rank approximation could be chosen considerably smaller (3–5 times) than the ensemble size  $N$  of the ensemble filter. However, the singular value decomposition becomes computationally the most expensive part of the reduced-rank algorithm for larger values of  $q$ .

In this paper we propose to combine the EnKF with the reduced-rank approach to reduce the statistical error of the ensemble filter. This is known as variance reduction, referring to the variance of the statistical error of the ensemble approach (see Hammersley and Handscomb 1964). The ensemble of the new filter algorithm now consists of two parts:  $q$  ensembles in the direction of the  $q$  leading eigenvalues of the covariance matrix and  $N$  randomly chosen ensembles. In the algorithm, only the projection of the random ensemble members orthogonal to the first ensemble members is used to obtain the filter gain and the state estimate. This partially orthogonal ensemble kalman filter (POEnKF) does not suffer from divergence problems because the reduced-rank approximation is embedded in an EnKF. The EnKF acts as a compensating mechanism for the truncation error. At the same time, POEnKF is much more accurate than the ensemble filter with ensemble size  $N + q$  because the leading eigenvectors of the covariance matrix are computed accurately using the full (extended) Kalman filter equations without statistical errors.

In section 2 we first show a simple example of variance reduction to illustrate the basic idea of our combined filter. In section 3 of this paper we summarize the ensemble kalman filter and the reduced-rank square root filter and introduce the partially orthogonal ensemble kalman filter algorithm and a few variants of this algorithm. We illustrate the performance of the various algorithms with an advection diffusion model application in section 4. Here, in order to compare the results of the various suboptimal filter algorithms with the exact Kalman filter, we concentrate our attention on linear problems.

## 2. Variance reduction: An illustrative example

Suppose we want to determine

$$E[f(\xi)], \quad (1)$$

where  $\xi$  is a random variable uniformly distributed on the interval  $[0, 1]$ . Using a Monte Carlo method we generate a sequence of uniformly distributed random numbers  $\xi_i$ ,  $i = 1, \dots, N$ , and compute

$$\bar{f} = \frac{1}{N} \sum_{i=1}^N f(\xi_i). \quad (2)$$

Here,  $\bar{f}$  is an estimator of  $E[f(\xi)]$ . The variance of this Monte Carlo estimator is (see Hammersley and Handscomb 1964)

$$\text{var}(\bar{f}) = \frac{1}{N} \int_0^1 \{f(\xi) - E[f(\xi)]\}^2 d\xi. \quad (3)$$

From this expression we see that the standard deviation of the statistical errors of the Monte Carlo method converge very slowly with the sample size ( $\approx 1/\sqrt{N}$ ).

Now let us suppose that we have an approximation  $\phi(\xi)$  of  $f(\xi)$  with known  $E[\phi(\xi)] = \Phi$ . Here,  $\phi(\xi)$  is also called a controlled variate. As a result we have, in this case,

$$E[f(\xi)] = \Phi + E[f(\xi) - \phi(\xi)]. \quad (4)$$

Now we can use as estimator of  $E[f(\xi)]$

$$\bar{f}_\phi = \Phi + \frac{1}{N} \sum_{i=1}^N [f(\xi_i) - \phi(\xi_i)] \quad (5)$$

with variance:

$$\begin{aligned} \text{var}\{\bar{f}_\phi\} &= \frac{1}{N} \int_0^1 [f(\xi) - \phi(\xi) - \{E[f(\xi)] - \Phi\}]^2 d\xi. \end{aligned} \quad (6)$$

If  $\phi(\xi)$  is reasonable approximation of  $f(\xi)$ ,  $\Phi$  will be a reasonable estimate of  $E[f(\xi)]$ . The Monte Carlo method, in this case, will only be used for estimating the remaining part  $E[f(\xi) - \phi(\xi)]$  with a variance of the error as given by (6). This variance is in general significantly smaller than the variance (3) of the original Monte Carlo approximation where  $\phi(\xi) = 0$ . Therefore  $\phi(\xi)$  is also often called a variance reductor. Variance reduction is attractive as long as we have an approximation  $\phi(\xi)$  of  $f(\xi)$  that is better than  $\phi(\xi) = 0$ . In most cases it will be easy to find an approximation that is better than doing nothing at all.

Let us for example take (see Fig. 1)

$$f(\xi) = 2\xi^2 - \xi^3,$$

and as approximation

$$\phi(\xi) = \xi.$$

In this case the variance (3) of the statistical error of the Monte Carlo method is  $0.1304/N$  while the variance (6) of the estimator (5) is only  $0.0304/N$ .

The basic idea just described can also be introduced for the ensemble Kalman filter. By computing an approximation of the filter problem using a reduced-rank filter, the ensemble Kalman filter needs only be used for

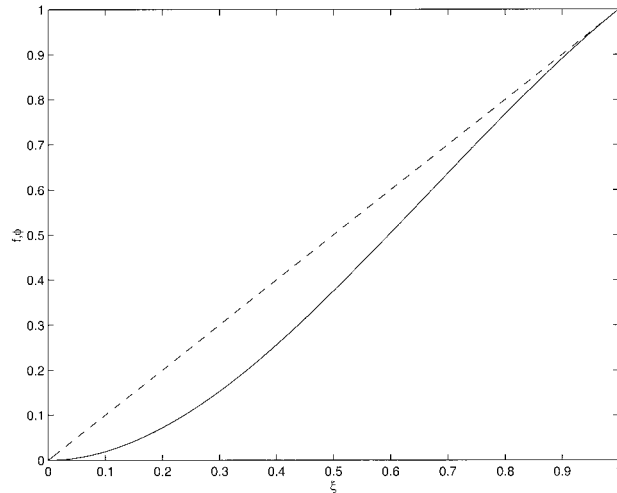


FIG. 1. Function  $f(\xi)$  (—) and its approximation  $\phi(\xi)$  (---) for  $0 \leq \xi \leq 1$ .

the remaining part of the problem. As a result, the statistical errors of the combined approach will be reduced significantly.

### 3. Kalman filtering

#### a. State space model

Suppose that modeling techniques have provided us with a nonlinear stochastic state space model of the form

$$\mathbf{x}^f(k+1) = \mathcal{M}[\mathbf{x}^f(k)] + \boldsymbol{\eta}(k), \quad (7)$$

and with observations

$$\mathbf{y}^o(k) = \mathbf{H}(k)\mathbf{x}^f(k) + \boldsymbol{\epsilon}(k). \quad (8)$$

Here  $\mathbf{x}^f(k)$  is the forecast of the system state at time  $k$  and  $\mathcal{M}[\mathbf{x}^f(k)]$  represents one time step of a numerical model. To model the inaccuracies of this model, a white system noise process  $\boldsymbol{\eta}(k)$  with covariance  $\mathbf{Q}(k)$  is introduced. Covariance  $\mathbf{Q}(k)$  is assumed to be of low rank so that it can be factorized as  $\mathbf{Q}(k) = \mathbf{Q}(k)^{1/2}[\mathbf{Q}(k)^{1/2}]^T$ , where  $\mathbf{Q}(k)^{1/2}$  is a matrix with relatively few columns. In the relation (8)  $\mathbf{y}^o(k)$  is the observations vector and  $\mathbf{H}(k)$  is the observation matrix that relates the available observations to the system state. The observation noise process  $\boldsymbol{\epsilon}(k)$  with covariance  $\mathbf{R}(k) = \mathbf{R}(k)^{1/2}[\mathbf{R}(k)^{1/2}]^T$  is introduced to model uncertainties in the observations.

#### b. Conventional Kalman filter for nonlinear systems

It is desired to combine the measurement taken from the actual system and modeled by relation (8) with the information provided by the system model (7) in order to obtain an optimal estimate of the system state. The optimal state estimate  $\mathbf{x}^a(k-1)$  at time  $k-1$  is forecast from observation time  $k-1$  to observation time  $k$  by the equations

$$\mathbf{x}^f(k) = \mathcal{M}[\mathbf{x}^a(k-1)] \quad (9)$$

$$\mathbf{P}^f(k) = \mathbf{M}(k-1)\mathbf{P}^a(k-1)\mathbf{M}(k-1)^T + \mathbf{Q}(k-1), \quad (10)$$

where

$$\mathbf{M}(k)_{ij} = \frac{\partial \mathcal{M}_i[\mathbf{x}^f(k)]}{\partial \mathbf{x}^f_j(k)} \quad (11)$$

represents the tangent linear model. At observation time  $k$  the observation  $\mathbf{y}^o(k)$  becomes available. The estimate is updated by the analysis equations

$$\mathbf{x}^a(k) = \mathbf{x}^f(k) + \mathbf{K}(k)[\mathbf{y}^o(k) - \mathbf{H}(k)\mathbf{x}^f(k)] \quad (12)$$

$$\mathbf{P}^a(k) = \mathbf{P}^f(k) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}^f(k), \quad (13)$$

where

$$\mathbf{K}(k) = \mathbf{P}^f(k)\mathbf{H}(k)^T[\mathbf{H}(k)\mathbf{P}^f(k)\mathbf{H}(k)^T + \mathbf{R}(k)]^{-1} \quad (14)$$

is the Kalman gain. The initial condition for the recursion is given by  $\mathbf{x}^a(0) = \mathbf{x}_0$  and  $\mathbf{P}^a(0) = \mathbf{P}_0$ .

#### c. Ensemble Kalman filter

The ensemble Kalman filter algorithm is based on a representation of the probability density of the state estimate by a finite number  $N$  of randomly generated system states.

The EnKF for the model (7)–(8) can be summarized as follows (see Burgers et al. 1998).

##### 1) INITIALIZATION

An ensemble of  $N$  states  $\xi_i^a(0)$  are generated to represent the uncertainty in  $\mathbf{x}_0$ ;

##### 2) FORECAST STEP

$$\xi_i^f(k) = \mathcal{M}[\xi_i^a(k-1)] + \boldsymbol{\eta}_i(k-1) \quad (15)$$

$$\mathbf{x}^f(k) = \frac{1}{N} \sum_{i=1}^N \xi_i^f(k) \quad (16)$$

$$\mathbf{E}^f(k) = [\xi_1^f(k) - \mathbf{x}^f(k), \dots, \xi_N^f(k) - \mathbf{x}^f(k)]; \quad (17)$$

##### 3) ANALYSIS STEP

$$\mathbf{P}^f(k) = \frac{1}{N-1} \mathbf{E}^f(k) \mathbf{E}^f(k)^T \quad (18)$$

$$\mathbf{K}(k) = \mathbf{P}^f(k)\mathbf{H}(k)^T[\mathbf{H}(k)\mathbf{P}^f(k)\mathbf{H}(k)^T + \mathbf{R}(k)]^{-1} \quad (19)$$

$$\xi_i^a(k) = \xi_i^f(k) + \mathbf{K}(k)[\mathbf{y}^o(k) - \mathbf{H}(k)\xi_i^f(k) + \boldsymbol{\epsilon}_i(k)], \quad (20)$$

where the ensemble of state vectors are generated with the realizations  $\boldsymbol{\eta}_i(k)$  and  $\boldsymbol{\epsilon}_i(k)$  of the noise processes  $\boldsymbol{\eta}(k)$  and  $\boldsymbol{\epsilon}(k)$ , respectively. Note that in the final implementation of the algorithm  $\mathbf{P}^f(k)$  need not actually be computed (see Evensen 1994).

For most practical problems the forecast equation (15) is computationally dominant (see Dee 1991). As a result the computational effort required for the EnKF is approximately  $N$  model simulations. The standard deviation of the errors in the state estimate are of a statistical nature and converge very slowly with the sample size ( $\approx 1/N$ ). This is one of the very few drawbacks of a Monte Carlo approach (see also section 2). Here it should be noted that for many atmospheric data assimilation problems the analysis step is also a very time consuming part of the algorithm (see Houtekamer and Mitchell 1998).

#### d. Reduced-rank square root filter

The reduced-rank square root (RRSQRT) filter algorithm (see Verlaan and Heemink 1997; Heemink et al. 1997) is based on a factorization of the covariance matrix  $\mathbf{P}$  of the state estimate according to  $\mathbf{P} = \mathbf{L}\mathbf{L}^T$ , where  $\mathbf{L}$  is a matrix with the  $q$  leading eigenvectors  $\mathbf{l}_i$  (scaled by the square root of the eigenvalues),  $i = 1, \dots, q$ , of  $\mathbf{P}$  as columns. The algorithm can be summarized as follows.

##### 1) INITIALIZATION

$$\mathbf{x}^a(0) = \mathbf{x}_0, \quad \mathbf{L}^a(0) = [\mathbf{l}_1^a(0), \dots, \mathbf{l}_q^a(0)]. \quad (21)$$

##### 2) FORECAST STEP

$$\mathbf{x}^f(k) = \mathcal{M}[\mathbf{x}^a(k-1)] \quad (22)$$

$$\mathbf{I}_i^f(k) = \frac{1}{\epsilon} \{ \mathcal{M}[\mathbf{x}^a(k-1) + \epsilon \mathbf{l}_i^a(k-1)] - \mathcal{M}[\mathbf{x}^a(k-1)] \}. \quad (23)$$

In Eq. (23)  $\epsilon$  represents a perturbation, often chosen close to 1:

$$\tilde{\mathbf{L}}^f(k) = [\mathbf{I}_1^f(k), \dots, \mathbf{I}_q^f(k), \mathbf{Q}(k-1)^{1/2}]. \quad (24)$$

In Eq. (24) a square root  $\tilde{\mathbf{L}}^f(k)$  of the forecast covariance is constructed by adding the columns of  $\mathbf{Q}(k-1)^{1/2}$ . However, the dimension of  $\tilde{\mathbf{L}}^f(k)$  has now increased. Therefore a new reduced-rank square root  $\mathbf{L}^f(k)$  is computed instead:

$$\mathbf{L}^f(k) = \mathbf{\Pi}^f(k) \tilde{\mathbf{L}}^f(k), \quad (25)$$

where  $\mathbf{\Pi}^f(k)$  is a projection onto the  $q$  leading eigenvectors of the matrix  $\tilde{\mathbf{L}}^f(k) \tilde{\mathbf{L}}^f(k)^T$ . Using the reduction step (25) we obtain an approximate square root  $\mathbf{L}^f(k)$  of  $\mathbf{P}^f(k)$  with only  $q$  columns.

##### 3) ANALYSIS STEP

$$\mathbf{P}^f(k) = \mathbf{L}^f(k) \mathbf{L}^f(k)^T \quad (26)$$

$$\mathbf{K}(k) = \mathbf{P}^f(k) \mathbf{H}(k)^T [\mathbf{H}(k) \mathbf{P}^f(k) \mathbf{H}(k)^T + \mathbf{R}(k)]^{-1} \quad (27)$$

$$\mathbf{x}^a(k) = \mathbf{x}^f(k) + \mathbf{K}(k) [\mathbf{y}^o(k) - \mathbf{H}(k) \mathbf{x}^f(k)] \quad (28)$$

$$\tilde{\mathbf{L}}^a(k) = \{ [\mathbf{I} - \mathbf{K}(k) \mathbf{H}(k)] \mathbf{L}^f(k), \mathbf{K}(k) \mathbf{R}(k)^{1/2} \} \quad (29)$$

$$\mathbf{L}^a(k) = \mathbf{\Pi}^a(k) \tilde{\mathbf{L}}^a(k), \quad (30)$$

where  $\mathbf{\Pi}^a(k)$  is a projection onto the  $q$  leading eigenvectors of the matrix  $\tilde{\mathbf{L}}^a(k) \tilde{\mathbf{L}}^a(k)^T$ . This reduction step is again introduced to reduce the number of columns in  $\mathbf{L}^a(k)$  to  $q$  in  $\tilde{\mathbf{L}}^a(k)$ . Note that the full covariance  $\mathbf{P}^f(k)$  need not to be computed (see Verlaan and Heemink 1997). The analysis step (29)–(30) is based on the Joseph form of the filter equations (see Maybeck 1979). This analysis step is not the most efficient procedure. Equations (29)–(30) are, however, more general than the analysis step proposed by Verlaan and Heemink (1997) and Heemink et al. (1997). Equations (29)–(30) hold for arbitrary filter gains  $\mathbf{K}(k)$  and not only for gain matrices satisfying Eq. (27). This becomes important if the RRSQRT algorithm is used as part of the POEnKF described in the next section.

For smaller values of  $q$ , Eq. (23) is for many practical problems computationally dominant, resulting in a computational effort of  $q + 1$  model simulations. The projections (25) and (30) require  $O(q^3)$  computations (see Heemink et al. 1997). As a result for very large  $q$  this part of the algorithm becomes time consuming too (see Cañizares 1999). Errors of the algorithm are caused by the fact that in the direction of the leading eigenvectors only two ensemble members are available and, as a result, the system dynamics is in essence linearized (see Heemink et al. 1997). Other errors are introduced by the representation of the covariance matrix by only the  $q$  leading eigenvectors. Because of the truncation, the covariance matrix is always underestimated. As a result the algorithm is sensitive to filter divergence problems. This problems can be avoided by choosing  $q$  relatively large, but this obviously reduces the computational efficiency.

#### e. Partially orthogonal ensemble Kalman filter

The ensemble Kalman filter and the reduced-rank square root filter are both of the square root type since both algorithms are formulated in terms of the square root of the covariance matrix [respectively  $\mathbf{E}^f(k)$  and  $\mathbf{L}^f(k)$ ]. Also, both algorithms are of the ensemble type. The ensemble filter is based on  $N$  randomly chosen ensemble members, while in the reduced-rank filter the ensemble members are chosen deterministically in the direction of the  $q$  leading eigenvectors. Because the two algorithms both have a very similar algorithmic structure, they can be integrated relatively easily.

The ensemble of the POEnKF consists of two parts:

The  $q$  leading eigenvectors  $\mathbf{l}_i$  of the covariance matrix plus  $N$  randomly chosen ensemble members  $\xi_i$ . The ensemble members  $\mathbf{l}_i$  are updated using an RRSQRT algorithm and  $\xi_i$  by using the EnKF algorithm. The two algorithms interact with each other at the analysis step. Here, only the information of the random ensemble members orthogonal to the  $q$  leading eigenvectors is used.

The POEnKF algorithm can be summarized as follows.

### 1) INITIALIZATION

$$[\mathbf{L}^a(0) \mathbf{E}^a(0)] = [\mathbf{l}_1^a(0), \dots, \mathbf{l}_q^a(0), \xi_1^a(0), \dots, \xi_N^a(0)] \quad (31)$$

where  $\mathbf{l}_i^a$  is the leading eigenvector of  $\mathbf{P}_0$  and  $\xi_i^a$  is generated randomly to represent the uncertainty in  $\mathbf{x}_0$ .

### 2) FORECAST STEP

forecast equations (22)–(25) of the RRSQRT algorithm,  
forecast equations (15)–(17) of the EnKF algorithm.

### 3) ANALYSIS STEP

In the analysis step we merge the information of the RRSQRT and the EnKF algorithms to obtain  $\mathbf{P}^f(k)$ . In the direction of the leading eigenvectors we use the results of the RRSQRT algorithm. In the other directions the RRSQRT algorithm does not give any information and we use the EnKF results. First we compute

$$\mathbf{E}^\perp(k) = \mathbf{\Pi}(k) \mathbf{E}^f(k), \quad (32)$$

where  $\mathbf{\Pi}^\perp(k)$  is a projection of the random ensemble members orthogonal to the first  $q$  ensemble members  $\mathbf{l}_i^f$ . Then we compute

$$\mathbf{P}^f(k) = \mathbf{L}^f(k) \mathbf{L}^f(k)^T + \frac{1}{N-1} \mathbf{E}^\perp(k) \mathbf{E}^\perp(k) \quad (33)$$

$$\mathbf{K}(k) = \mathbf{P}^f(k) \mathbf{H}(k)^T [\mathbf{H}(k) \mathbf{P}^f(k) \mathbf{H}(k)^T + \mathbf{R}(k)]^{-1}, \quad (34)$$

analysis Eq. (20) of the EnKF algorithm for the ensemble  $\mathbf{E}^f(k)$ ,

analysis Eqs. (28)–(30) of the RRSQRT algorithm.

For small values of  $q$  the time propagation equations (15) and (23) for the ensemble is computationally dominating. As a result for most practical problems the computational effort for the POEnKF is approximately  $N + q$  times the effort required for one model simulation. By integrating the ensemble Kalman filter and the reduced-rank square root filter the best of both are combined. The reduced-rank part acts as a variance reductor for the ensemble filter reducing the statistical errors of

this Monte Carlo approach significantly. On the other hand, by embedding the reduced-rank filter in an Ensemble Kalman filter the covariance is not underestimated, eliminating the filter divergence problems of the reduced-rank approach (also for very small numbers of  $q$ ). As a result  $q$  can be chosen on the basis of efficiency arguments and not for stabilizing the filter algorithm.

Many generalizations and modifications of the POEnKF exists. One important modification to make the POEnKF more robust for strongly nonlinear problems is to take into account the information of the random ensemble members in the direction of the  $q$  leading eigenvectors of the covariance matrix. If

$$\mathbf{E}''(k) = \mathbf{\Pi}''(k) \mathbf{E}^f(k) \quad (35)$$

is the projection of the random ensemble members onto the first  $q$  ensemble members  $\mathbf{l}_i$ , then

$$\frac{1}{N-1} \mathbf{E}''(k) \mathbf{E}''(k)^T \quad (36)$$

is the ensemble Kalman filter approximation of the reduced-rank covariance matrix with rank  $q$ . Instead of Eq. (33), where this information is completely ignored, it is also possible to use the equation

$$\begin{aligned} \mathbf{P}^f(k) = & \gamma \mathbf{L}^f(k) \mathbf{L}^f(k)^T + \frac{1-\gamma}{N-1} \mathbf{E}''(k) \mathbf{E}''(k)^T \\ & + \frac{1}{N-1} \mathbf{E}^\perp(k) \mathbf{E}^\perp(k)^T, \end{aligned} \quad (37)$$

where  $0 \leq \gamma \leq 1$  is a weighting coefficient. The first term of the right-hand side suffers from errors caused by the approximation of the nonlinear dynamic of the system in the forecast equation, the second term is dominated by the statistical noise of the Monte Carlo method. For most practical problems the errors of the first term are relatively small and  $\gamma = 1$  is a good choice. However, for very nonlinear data assimilation problems it may be better to choose  $\gamma$  smaller.

### f. Complementary orthogonal subspace filter for efficient ensembles

The POEnKF described in the previous section uses a RRSQRT time update for a small number of directions that correspond to the largest eigenvalues of the forecast error covariance. A separate ensemble update is used for the remaining space. However, the covariance for this orthogonal subspace is obtained by first generating an ensemble for the whole state space and subsequently removing all contributions in the directions with the largest variance. Since the contribution of the ensemble is ignored in the subspace spanned by modes of the RRSQRT part of the POEnKF, it seems more efficient to remove those contributions from the ensemble altogether. This can be achieved with a simple modification of the POEnKF, which will be referred to hereafter as



the complementary orthogonal subspace filter for efficient ensembles (COFFEE) algorithm.

The COFFEE algorithm can be summarized as follows.

### 1) INITIALIZATION

$$\begin{aligned} & [\mathbf{L}^a(0) \mathbf{E}^a(0)] \\ &= [\mathbf{I}_1^a(0), \dots, \mathbf{I}_q^a(0), \boldsymbol{\xi}_1^a(0), \dots, \boldsymbol{\xi}_N^a(0)], \quad (38) \end{aligned}$$

where  $\mathbf{I}_q^a$  again denotes the  $q$  leading eigenvectors of  $\mathbf{P}_0$ . However, the random ensemble members are now generated only to represent the covariance  $\mathbf{P}_0 - \mathbf{L}^a(0)\mathbf{L}^a(0)^T$ . This is the part of the covariance that is not represented by the leading eigenvectors.

### 2) FORECAST STEP

$\mathbf{L}^a$  is updated using the RRSQRT update (22)–(25) and the ensemble  $\boldsymbol{\xi}_i^a$  is updated using (15)–(17). However, a random ensemble is now generated with an additional covariance equal to the covariance of the truncated part in the RRSQRT half, that is, (17) changes to

$$\begin{aligned} \mathbf{E}^f(k) &= [\boldsymbol{\xi}_1^f(k) - \mathbf{x}^f(k) + \boldsymbol{\eta}_1(k), \dots, \boldsymbol{\xi}_N^f(k) - \mathbf{x}^f(k) \\ &\quad + \boldsymbol{\eta}_N(k)], \quad (39) \end{aligned}$$

with  $E[\boldsymbol{\eta}_i(k)\boldsymbol{\eta}_i(k)'] = [\mathbf{I} - \mathbf{\Pi}^f(k)]\mathbf{L}^f(k)\mathbf{L}^f(k)^T[\mathbf{I} - \mathbf{\Pi}^f(k)]^T$ . In the RRSQRT forecast step, only the  $q$  leading eigenvectors are taken into account. The other eigenvectors are neglected. In the COFFEE forecast step, the part of the covariance that is truncated in the RRSQRT forecast step is also represented by the random ensembles.

### 3) ANALYSIS STEP

The measurement update is almost identical to that of the POEnKF, except that no projection of the ensemble is needed, that is, (32) can be omitted.

## 4. Application

### a. Model and experimental setup

Before using the new algorithms for real-life atmospheric chemistry data assimilations problems, we define in this paper first a test problem. Consider the 2D advection diffusion equation

$$\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} = \nu \frac{\partial^2 c}{\partial x^2} + \nu \frac{\partial^2 c}{\partial y^2} + S, \quad (40)$$

with a square domain  $[0, L] \times [0, L]$  and zero initial conditions. Here,  $c$  is the concentration,  $[u, v]$  is the velocity field,  $\nu$  is the dispersion coefficient, and  $S$  represents the source terms. The concentration at the boundary is zero for inflow. A backward Lagrangian scheme is used to discretize these equations on a 30  $\times$

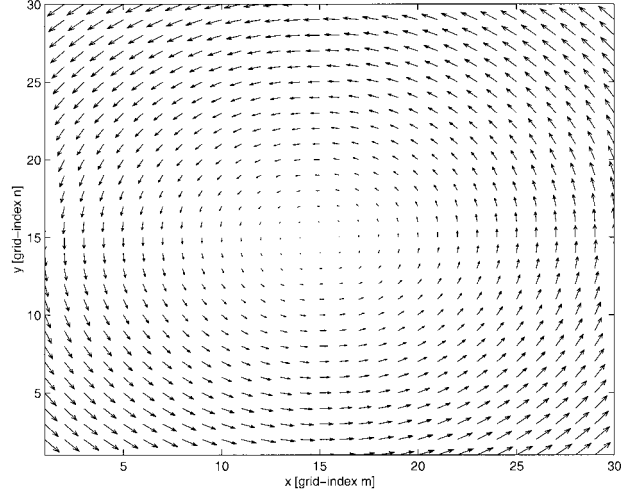


FIG. 2. Velocity field.

30 grid. Preliminary experiments indicated that the numerical diffusion was still so large that the physical diffusion terms could be omitted. For all experiments hereafter the velocity field was assumed to be known and constant in time. Figure 2 shows the velocity field that is similar to that of the well-known Molenkamp test.

In order to compare the algorithms proposed to other existing algorithms some twin experiments were carried out. A reference solution was generated by inserting constant emissions at grid cells  $\{(5, 8), (7, 8), (15, 5), (3, 18), (20, 15)\}$ . The increase of concentration per time step for these location was  $\{0.2, 0.5, 0.4, 0.2, 0.25\}$ , respectively.

Observations were generated from simulated true concentrations, which were computed by adding fluctuations to the mean emissions. For this purpose an AR(1) process was used, that is, the fluctuations to the emissions per time step were computed according to

$$\tilde{\eta}_j(k+1) = \alpha_j \tilde{\eta}_j(k) + \eta_j(k), \quad (41)$$

with  $\eta_j$  independent Gaussian white noise processes with  $E[\eta_j(k)] = 0$  and  $\text{var}[\eta_j(k)] = 1$ . The index  $j$  refers to source locations  $\{(m, n)_1 \dots (m, n)_p\} = \{(4, 11), (13, 3), (26, 15), (15, 10), (23, 2), (11, 9), (15, 20), (23, 10), (5, 25)\}$ . The decays per time step are  $\{\alpha_1 \dots \alpha_p\} = \{0.9, 0.8, 0.8, 0.9, 0.9\}$ . Negative emissions are truncated, which creates some nonlinearity in the data-assimilation problem. Finally, white observational noise with variance 0.01 is added to the true concentrations.

### b. Results of the suboptimal Kalman filter algorithms

In evaluating the filter algorithms we compute the rms error

$$\text{RMS} := \sqrt{\frac{1}{M^2 K} \sum_{m,n,k} [c_{m,n}(k) - \hat{c}_{m,n}(k)]^2}, \quad (42)$$

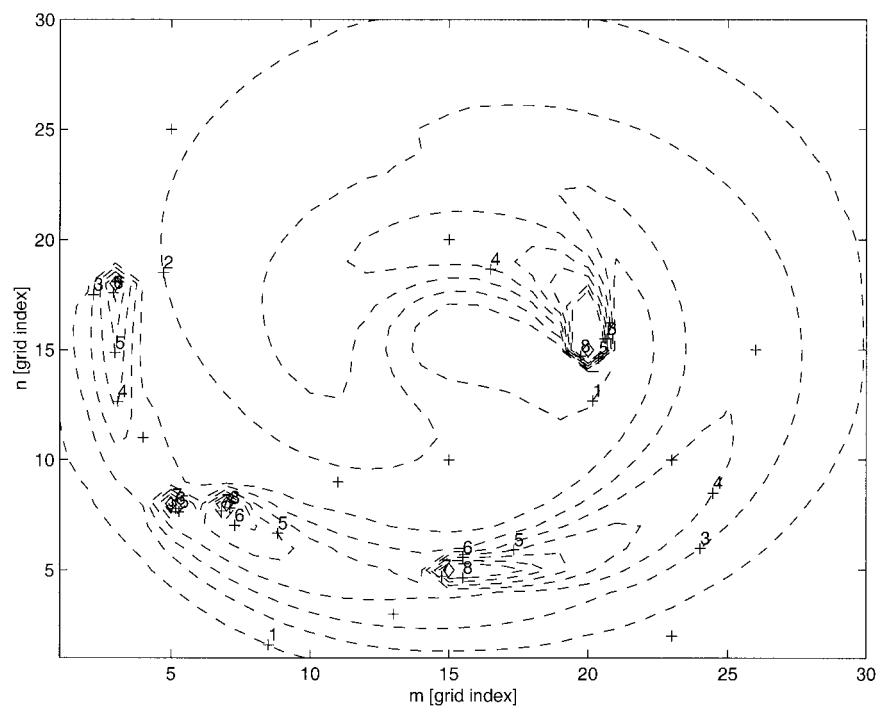


FIG. 3. Reference concentration field after  $k = 100$  time steps. Contours at shown with increment 1 for reference concentrations.

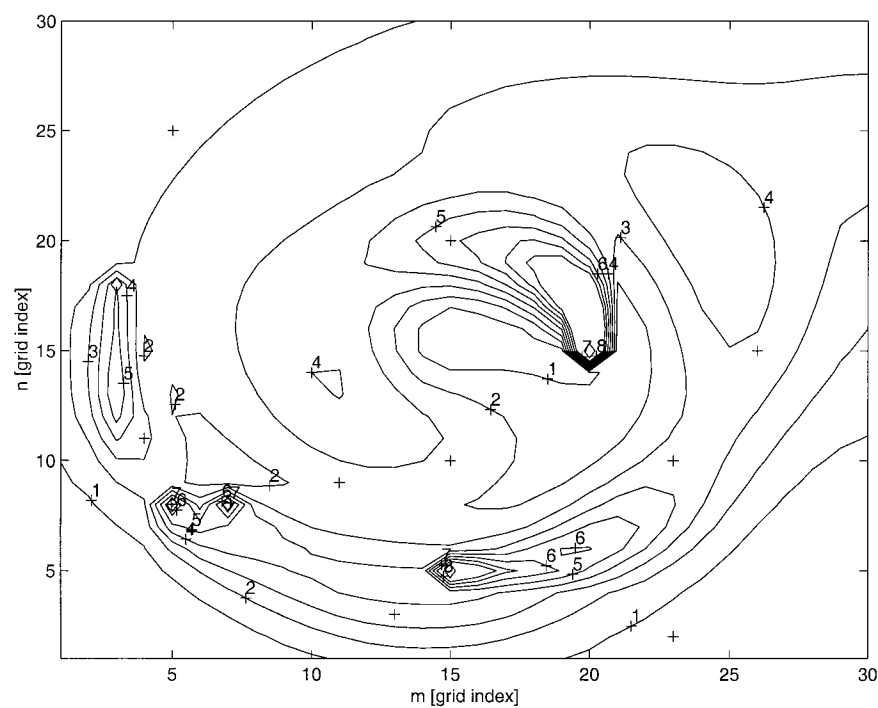


FIG. 4. True concentration field after  $k = 100$  time steps. Contours at shown with increment 1 for reference concentrations.

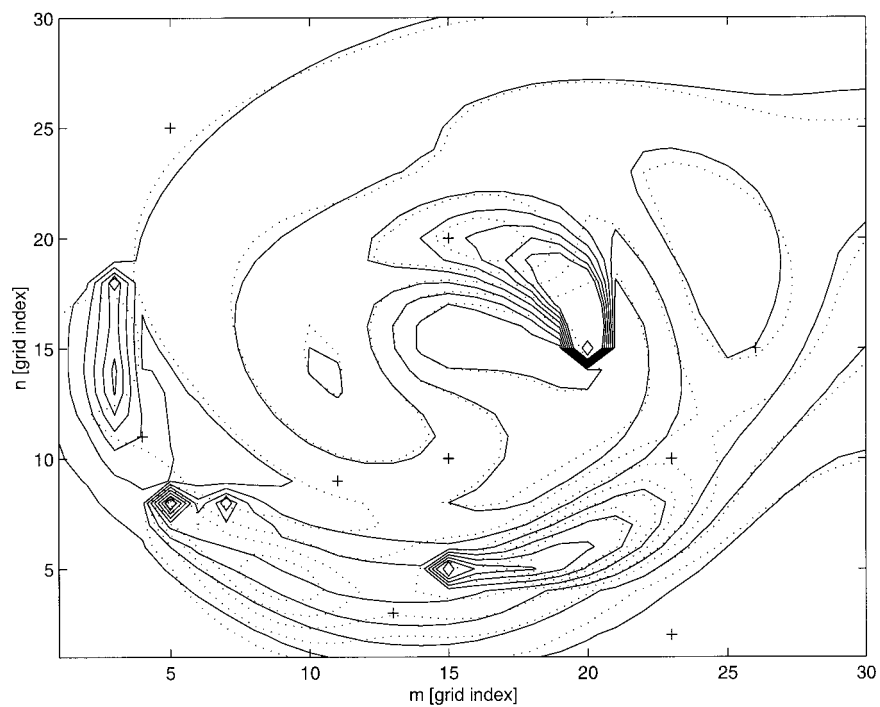


FIG. 5. Concentrations computed with the EnKF with  $N = 20$  at  $k = 100$ . Contours at shown with increment 1 for true concentrations (—) and filter solution ( $\cdots$ ).

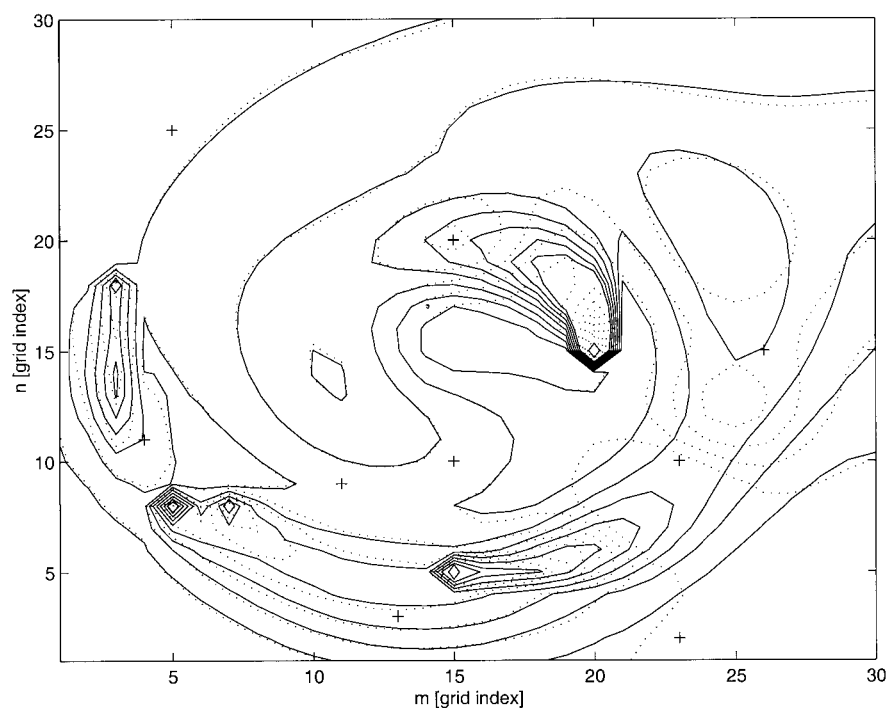


FIG. 6. Concentrations computed with the RRSQRT filter with  $N = 20$  at  $k = 100$ . Contours at shown with increment 1 for true concentrations (—) and filter solution ( $\cdots$ ).



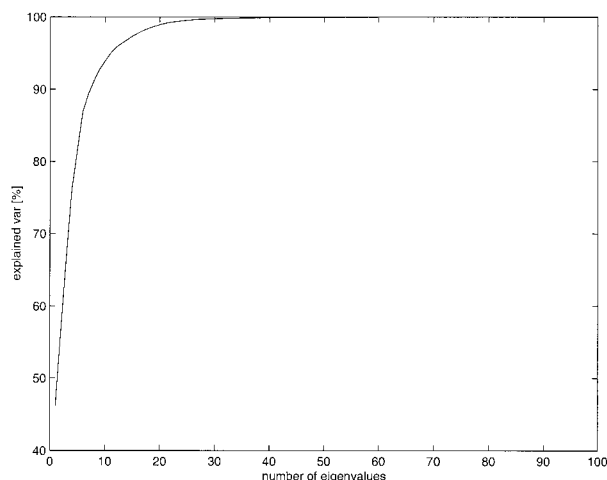


FIG. 7. Eigenvalues of the error covariance matrix  $P(100|100)$ . Computed with the RRSQRT filter with  $q = 50$ .

where  $c_{m,n}(k)$  are the exact generated concentrations and  $\hat{c}_{m,n}(k)$  are the estimates computed,  $M$  is the number of grid points in one direction, and  $K$  is the number of time steps.

In Figs. 3 and 4 the concentration fields of the truth run and the reference run are shown after  $K = 100$  time steps. The + signs indicate measurement locations and the diamond signs the locations of the emissions. It can be seen clearly that the true fields are perturbed with

time-varying fluctuations, while the reference solutions only contains a steady emission that is advected and spreading smoothly. The rms difference between both runs is 0.665 without any data assimilation.

In Figs. 5 and 6 we show the results for the ensemble Kalman filter and the reduced-rank square root filter at the final time  $k = 100$  for 20 ensemble members and 20 modes, respectively. These settings correspond to approximately the computational requirements for 20 model runs. For these settings neither filter performs well. Their respective rms errors are 0.838 and 0.797, which is worse than without data assimilation. The contours clearly show that both algorithms create spurious correlations if not fully converged. This result is consistent with the findings of Houtekamer and Mitchel (1998), who found that there tend to be spurious correlations at long distances when ensemble sizes are small. They introduced a data-selection procedure to reduce these effects. Although 20 eigenvalues explain almost 98% of the error covariance, both the EnKF and the RRSQRT filter still show divergence problems in the sense that the rms error using data assimilation is larger than the error without any data assimilation (see also Fig. 10). See Fig. 7 for the variance explained by the leading eigenvectors of the analysis error covariance at the final time  $k = 100$ , as computed with the RRSQRT filter with 100 modes.

In Figs. 8 and 9 the results for the POEnKF and the COFFEE are shown at the final time. In both computations 15 modes and an ensemble with  $N = 5$  members

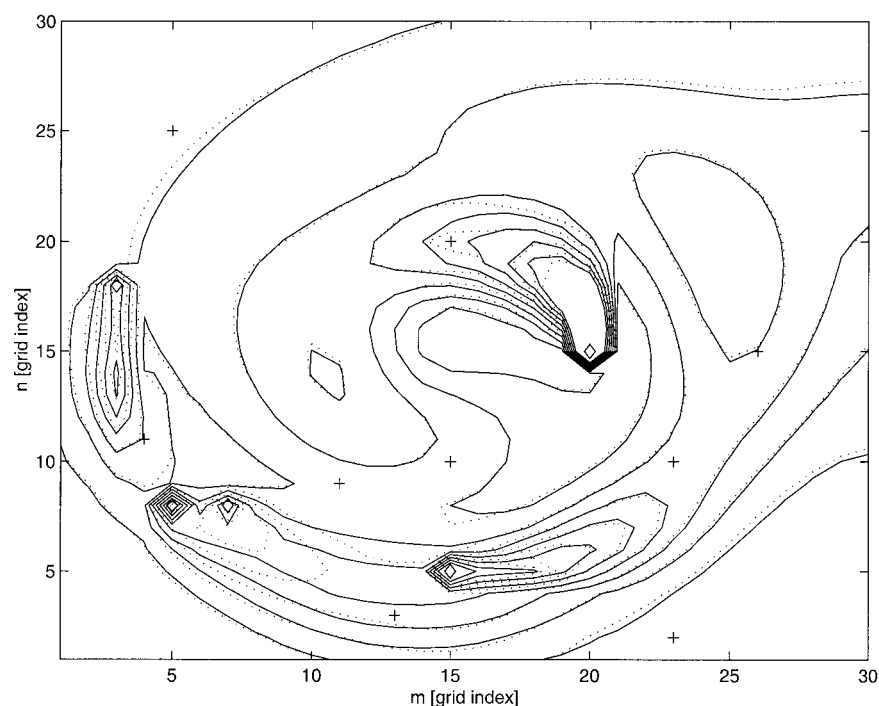


FIG. 8. Concentrations computed with the POEnKF with  $q + N = 15 + 5$  at  $k = 100$ . Contours are shown with increment 1 for true concentrations (—) and filter solution (· · ·).

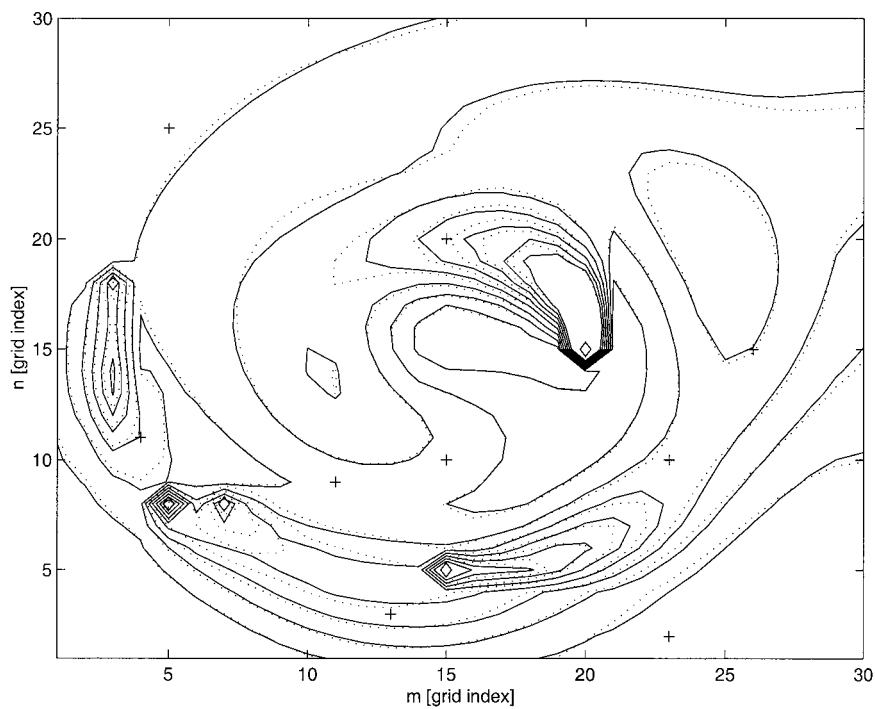


FIG. 9. Concentrations computed with the COFFEE KF with  $q + N = 15 + 5$  at  $k = 100$ . Contours are shown with increment 1 for true concentrations (—) and filter solution (· · ·).

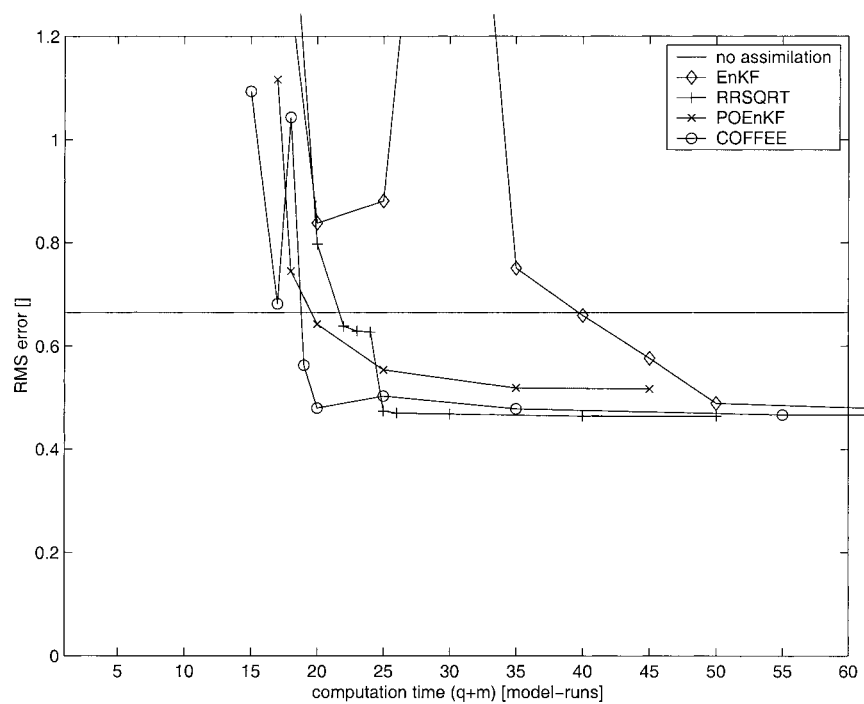


FIG. 10. Rms errors vs computation times for the suboptimal Kalman filter algorithms.

was used. Although the computational requirements are approximately the same as in the previous experiments these filters do not diverge and actually improve on the reference solution. The rms's are 0.642 and 0.480, respectively. This also shows that the reduced variance in the COFFEE algorithm as compared to the POEnKF does indeed reduce the errors even further in this case. In other experiments not shown here we also found that of the various variants that are possible for the COFFEE algorithm, it is most efficient to project the Kalman gain onto the subspace spanned by the modes of the RRSQRT part of the algorithm, that is, to effectively ignore correlation computed from the EnKF part of the algorithm but to include their variance in the innovation error covariance.

Figure 10 shows the convergence of the four sub-optimal schemes used in this paper. For the POEnKF and the COFFEE algorithm the number of random ensemble members was kept fixed at  $N = 5$ . We observed during the experiments that for a small number of modes or ensemble members the rms error are sensitive to the random numbers used in the computation. To reduce this random effect on the results the same set of random numbers was used for all the experiments, which made the results more comparable. However, the seemingly random behavior for small ensembles (or modes) is still influenced by the set of random numbers used. When comparing the algorithms, it can be seen that both the EnKF and the RRSQRT algorithms seem to diverge for small ensemble size. The full (extended) Kalman filter produces an rms error of 0.46.

Both the POEnKF and the COFFEE algorithm improve the convergence for small ensemble sizes, where the COFFEE variant is slightly more efficient. However, if the number of modes is increased the random part of these algorithms seems to hamper quick convergence of the RRSQRT part of the algorithm. In this experiment accurate results are obtained with the COFFEE algorithm at only 40% of the computation time compared to the EnKF ( $q = 20$  and  $N = 50$ ).

## 5. Conclusions

In this paper we propose to use a reduced-rank square root Kalman filter as a variance reductor for the ensemble Kalman filter to reduce the statistical errors of this Monte Carlo approach. The resulting partially orthogonal ensemble filter and its variants combine the best of both. It is less sensitive to divergence problems and

much faster than both the reduced-rank approach and the ensemble Kalman filter. Next, the POEnKF and COFFEE algorithms will be applied to real-life, nonlinear data assimilation problems in atmospheric chemistry and shallow water flow forecasting. Furthermore, we will also study the performance of the new algorithms in nonlinear, unstable systems.

## REFERENCES

- Burgers, G., P. J. Van Leeuwen, and G. Evensen 1998: Analysis scheme in the ensemble Kalman filter. *Mon. Wea. Rev.*, **126**, 1719–1724.
- Cañizares, R., 1999: On the application of data assimilation in regional coastal models. Ph.D. thesis, Delft University of Technology, Delft Netherlands, 133 pp.
- Cohn, S. E., and R. Todling, 1995: Approximate Kalman filters for unstable dynamics. *Second Int. Symp. on Assimilation of Observations in Meteorology and Oceanography*, Tokyo, Japan, WMO, 241–246.
- , and —, 1996: Approximate data assimilation schemes for stable and unstable dynamics. *J. Meteor. Soc. Japan*, **74**, 63–75.
- Dee, D. P., 1991: Simplification of the Kalman filter for meteorological data assimilation. *Quart. J. Roy. Meteor. Soc.*, **117**, 365–384.
- Evensen, E., 1994: Sequential data assimilation with a nonlinear QG model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, **99**, 10 143–10 162.
- Evensen, G., and P. J. Van Leeuwen, 1996: Assimilation of geosat altimeter data for the agulhas current using the Ensemble Kalman filter with a quasigeostrophic model. *Mon. Wea. Rev.*, **124**, 85–96.
- Ghil, M., and P. Malanotte-Rizzoli, 1991: Data assimilation in meteorology and oceanography. *Advances in Geophysics*, vol. 33, Academic Press, 141–266.
- Hammersley, J., and D. Handscomb, 1964: *Monte Carlo Methods*. Wiley and Sons.
- Heemink, A. W., K. Bolding, and M. Verlaan, 1997: Storm surge forecasting using Kalman filtering. *J. Meteor. Soc. Japan*, **75**, 305–318.
- Houtekamer, P. L., and A. L. Mitchell, 1998: Data assimilation using ensemble Kalman filter technique. *Mon. Wea. Rev.*, **126**, 796–811.
- Lermusiaux, P. F. J., 1997: Error subspace data assimilation methods for ocean field estimation: Theory, validation and applications. Ph. D. thesis, Harvard University, 402 pp.
- Maybeck, P. S., 1979: *Stochastic Models, Estimation and Control*. Vol. 141–1, *Mathematics in Science and Engineering*, Academic Press, 423 pp.
- Pham, D., J. Verron, and M. Roubaud, 1998: A singular evolutive extended Kalman filter for data assimilation in oceanography. *J. Mar. Syst.*, **16**, 323–340.
- Verlaan, M., and A. W. Heemink, 1995: Data assimilation schemes for non-linear shallow water flow models. *Proc. Second Int. Symp. on Assimilation of Observations*, Tokyo, Japan, WMO, 247–252.
- , and —, 1997: Tidal flow forecasting using reduced-rank square root filters. *Stochastic Hydro. Hydraul.*, **11**, 349–368.