Ocean Dynamics (2003) 53: 343–367
DOI 10.1007/s10236-003-0036-9

Geir Evensen

The Ensemble Kalman Filter: theoretical formulation and practical implementation

Received: 16 December 2002 / Accepted: 7 May 2003

Abstract The purpose of this paper is to provide a comprehensive presentation and interpretation of the Ensemble Kalman Filter (EnKF) and its numerical implementation. The EnKF has a large user group, and numerous publications have discussed applications and theoretical aspects of it. This paper reviews the important results from these studies and also presents new ideas and alternative interpretations which further explain the success of the EnKF. In addition to providing the theoretical framework needed for using the EnKF, there is also a focus on the algorithmic formulation and optimal numerical implementation. A program listing is given for some of the key subroutines. The paper also touches upon specific issues such as the use of nonlinear measurements, in situ profiles of temperature and salinity, and data which are available with high frequency in time. An ensemble based optimal interpolation (EnOI) scheme is presented as a cost-effective approach which may serve as an alternative to the EnKF in some applications. A fairly extensive discussion is devoted to the use of time correlated model errors and the estimation of model bias.

Keywords Data assimilation · Ensemble Kalman Filter

1 Introduction

The Ensemble Kalman Filter has been examined and applied in a number of studies since it was first introduced by Evensen (1994b). It has gained popularity because of its simple conceptual formulation and relative ease of implementation, e.g., it requires no derivation of a tangent linear operator or adjoint equations, and no integrations backward in time. Further, the computational requirements are affordable and comparable with other popular sophisticated assimilation methods such as the representer method by Bennett (1992); Bennett et al. (1993, 1996); Bennett and Chua (1994) and the 4DVAR method which has been much studied by the meteorological community (see, e.g., Talagrand and Courtier 1997, 1987; Courtier and Talagrand 1987; Courtier et al. 1994).

This paper gives a comprehensive presentation of the Ensemble Kalman Filter (EnKF), and it may serve as an EnKF reference document. For a user of the EnKF it provides citations from hopefully all previous publications where the EnKF has been examined or used. It also provides a detailed presentation of the method in terms of both theoretical aspects and practical implementation. For experienced EnKF users it will provide a better understanding of the EnKF through the presentation of a new and alternative interpretation and implementation of the analysis scheme.

In the next section, an overview is given of previous works involving the EnKF. Further, in Section 3, an overview of the theoretical formulation of the EnKF will be given. Thereafter the focus will be on implementation issues, starting with the generation of the initial ensemble in Section 4.1 and the stochastic integration of the ensemble members in Section 4.2. The major discussion in this paper relates to the EnKF analysis scheme, which is given in Section 4.3. Section 5 discusses particular aspects of the numerical implementation. Appendix A presents an approach for examining the consistency of the EnKF based on comparisons of innovations and predicted error statistics. In Appendix B an optimal interpolation algorithm is presented. It uses a stationary ensemble but is otherwise similar to the EnKF, and it can thus be denoted Ensemble Optimal Interpolation (EnOI). In Appendix C we have given an algorithm which is currently used for assimilation of observations of subsurface quantities. In Appendix D the Ensemble

Responsible Editor: Jörg-Olaf Wolff

Geir Evensen
Nansen Environmental and Remote Sensing Center,
Bergen Norway Edvard Griegsvei 3A,
5059 Solheimsviken, Norway
e-mail: Geir.Evensen@nersc.no

Kalman Smoother (EnKS) is presented in terms of the terminology developed in this paper. It is illustrated how the smoother solution can be very efficiently computed as a reanalysis following the use of the EnKF. In Appendix E we have reviewed and detailed the presentation of the algorithm used for the generation of pseudorandom fields. Finally, in Appendix F an example is given illustrating the EnKF and EnKS with a simple stochastic scalar model. This illustrates the use of time-correlated model errors and how these can be estimated. The use of the EnKF and EnKS for estimation of model bias is given in Appendix G.

2 Chronology of ensemble assimilation developments

This section attempts to provide a complete overview of the developments and applications related to the EnKF. In addition, it also points to other recently proposed ensemble-based methods and some smoother applications.

2.1 Applications of the EnKF

Applications involving the EnKF are numerous and include the initial work by Evensen (1994b) and an additional example in Evensen (1994a), which showed that the EnKF resolved the closure problems reported from applications of the Extended Kalman Filter (EKF).

An application with assimilation of altimeter data for the Agulhas region was discussed in Evensen and van Leeuwen (1996) and later in a comparison with the Ensemble Smoother (ES) by van Leeuwen and Evensen (1996).

An example with the Lorenz equations was presented by Evensen (1997), where it was shown that the EnKF could track the phase transitions and find a consistent solution with realistic error estimates even for such a chaotic and nonlinear model.

Burgers et al. (1998) reviewed and clarified some points related to the perturbation of measurements in the analysis scheme, and also gave a nice interpretation supporting the use of the ensemble mean as the best estimate.

Houtekamer and Mitchell (1998) introduced a variant of the EnKF where two ensembles of model states are integrated forward in time, and statistics from one ensemble are used to update the other. The use of two ensembles was motivated by claiming that this would reduce possible inbreeding in the analysis. This has, however, led to some dispute, discussed in the comment by van Leeuwen (1999a) and the reply by Houtekamer and Mitchell (1999).

Miller et al. (1999) included the EnKF in a comparison with nonlinear filters and the Extended Kalman Filter, and concluded that it performed well, but could be beaten by a nonlinear and more expensive filter in difficult cases where the ensemble mean is not a good estimator.

Madsen and Cañizares (1999) compared the EnKF and the reduced rank square root implementation of the Extended Kalman filter with a 2-D storm surge model. This is a weakly nonlinear problem, and good agreement was found between the EnKF and the extended Kalman filter implementation.

Echevin et al. (2000) studied the EnKF with a coastal version of the Princeton Ocean Model and focused in particular on the horizontal and vertical structure of multivariate covariance functions from sea-surface height. It was concluded that the EnKF could capture anisotropic covariance functions resulting from the impact of coastlines and coastal dynamics, and had a particular advantage over simpler methodologies in such areas.

Evensen and van Leeuwen (2000) rederived the EnKF as a suboptimal solver for the general Bayesian problem of finding the posterior distribution given densities for the model prediction and the observations. From this formulation the general filter, could be derived and the EnKF could be shown to be a suboptimal solver of the general filter, where the prior densities are assumed to be Gaussian distributed.

Hamill and Snyder (2000) constructed a hybrid assimilation scheme by combining 3DVAR and the EnKF. The estimate is computed using the 3DVAR algorithm but the background covariance is a weighted average of the time evolving EnKF error covariance and the constant 3DVAR error covariance. A conclusion was that with increasing ensemble size the best results were found with larger weight on the EnKF error covariance.

Hamill et al. (2000) report from working groups in a workshop on ensemble methods.

Keppenne (2000) implemented the EnKF with a two-layer shallow-water model and examined the method in twin experiments assimilating synthetic altimetry data. He focused on the numerical implementation on parallel computers with distributed memory, and found the approach efficient for such systems. He also examined the impact of ensemble size and concluded that realistic solutions could be found using a modest ensemble size.

Mitchell and Houtekamer (2000) introduced an adaptive formulation of the EnKF, where the model error parameterization was updated by incorporating information from the innovations during the integration.

Park and Kaneko (2000) presented an experiment where the EnKF was used to assimilate acoustic tomography data into a barotropic ocean model.

van Loon et al. (2000) used the EnKF for assimilation of ozone data into an atmospheric transport chemistry model.

Grønnevik and Evensen (2001) examined the EnKF for use in fish stock assessment, and also compared it with the Ensemble Smoother (ES) by van Leeuwen and Evensen (1996) and the more recent Ensemble Kalman Smoother (EnKS) by Evensen and van Leeuwen (2000).

Heemink et al. (2001) have been examining different approaches which combine ideas from RRSQRT filtering and the EnKF to derive computationally more efficient methods.

Houtekamer and Mitchell (2001) have continued the examination of the two-ensemble approach and introduced a technique for computing the global EnKF analysis in the case with many observations, and also a method for filtering of eventual long range spurious correlations caused by a limited ensemble size. As will be seen below, the current paper presents a much more efficient way to compute the global analysis and also argues against filtering of covariances.

Pham (2001) reexamined the EnKF in an application with the Lorenz attractor and compared results with those obtained from different versions of the Singular Evolutive Extended Kalman (SEEK) filter and a particle filter. Ensembles with very few members were used and this favored methods like the SEEK where the "ensemble" of EOFs is selected as a best-possible representation of the model attractor.

Verlaan and Heemink (2001) applied the RRSQRT and EnKF filters in test examples with the purpose of classifying and defining a measure of the degree of nonlinearity of the model dynamics. Such an estimate may have an impact on the choice of assimilation method.

Hansen and Smith (2001) proposed a method for producing analysis ensembles based on integrated use of the 4DVAR method and the EnKF. A probabilistic approach was used and lead to high numerical cost, but an improved estimate could be found compared to 4DVAR and the EnKF used separately.

Hamill et al. (2001) examined the impact of ensemble size on noise in distant covariances. They evaluated the impact of using an inflation factor, as introduced by Anderson and Anderson (1999), and also the use of a Schur product of the covariance with a correlation function to localize the background covariances, as previously discussed by Mitchell (2001). The inflation factor is used to replace the forecast ensemble according to:

$$\psi_j = \rho(\psi_j - \bar{\psi}) + \bar{\psi}, \tag{1}$$

with ρ slightly greater than 1 (typically 1.01). The purpose is to account for a slight underrepresentation of variance due to the use of a small ensemble.

Bishop et al. (2001) used an implementation of the EnKF in an observation system simulation experiment. Ensemble-predicted error statistics were used to determine the optimal configuration of future targeted observations. The application typically looked at a case where additional targeted measurements could be deployed over the next few days and the deployment could be optimized to minimize the forecast errors in a selected region. The methodology was named Ensemble Transform Kalman Filter and it was further examined by Majumdar et al. (2001).

Reichle et al. (2002) give a nice discussion of the EnKF in relation to the optimal representer solution. They find good convergence of the EnKF toward the representer solution, with the difference being caused by the Gaussian assumptions used in the EnKF at analysis steps. These are avoided in the representer method, which solves for the maximum-likelihood smoother estimate.

Bertino et al. (2002) applied the EnKF and the Reduced Rank Square Root (RRSQRT) filter with a model for the Odra estuary. The two methods were compared and used to assimilate real observations to assess the potential for operational forecasting in the lagoon. This is a relatively linear model and the EnKF and the RRSQRT filter provided similar results.

Eknes and Evensen (2002) examined the EnKF with a 1-D three component marine ecosystem model with focus on sensitivity to the characteristics of the assimilated measurements and the ensemble size. It was found that the EnKF could handle strong nonlinearities and instabilities which occur during the spring bloom.

Allen et al. (2002) take the Eknes and Evensen (2002) work one step further by applying the method with a 1-D version of ERSEM for a site in the Mediterranean Sea. They showed that even with such a complex model it is possible to find an improved estimate by assimilating in situ data into the model.

Haugen and Evensen (2002) applied the EnKF to assimilate sea-level anomalies and sea-surface temperature data into a version of the Miami Isopycnic Coordinate Ocean Model (MICOM) by Bleck et al. (1992) for the Indian Ocean. The paper provided an analysis of regionally dependent covariance functions in the tropics and subtropics and also the multivariate impact of assimilating satellite observations.

Mitchell et al. (2002) examined the EnKF with a global atmospheric general circulation model with simulated data resembling realistic operational observations. They assimilated 80 000 observations a day. The system was examined with respect to required ensemble size, and the effect of localization (local analysis at a grid point using only nearby measurements). It was found that severe localization could lead to imbalance, but with large enough ratio of influence for the measurements, this was not a problem, and no digital filtering was required. In the experiments they also included model errors and demonstrated the importance of this to avoid filter divergence. This work is a significant step forward and it shows promising results with respect to using the EnKF with atmospheric forecast models.

Crow and Wood (2003) demonstrated that the EnKF is an effective and a computationally competitive strategy for the assimilation of remotely sensed brightness temperature measurements into land-surface models.

Brusdal et al. (2003) discussed a similar application to Haugen et al. (2002), but focused on the North Atlantic. In addition, this paper included an extensive comparison of the theoretical background of the EnKF,

EnKS, and the SEEK filter, and also compared results from these methods.

Natvik and Evensen (2003a,b) presented the first realistic 3-D application of the EnKF with a marine ecosystem model. These papers proved the feasibility of assimilating SeaWiFS ocean color data to control the evolution of a marine ecosystem model. In addition, several diagnostic methods were introduced which can be used to examine the statistical and other properties of the ensemble.

Keppenne and Rienecker (2003) implemented a massively parallel version of the EnKF with the Poseidon isopycnic coordinate ocean model for the tropical Pacific. They demonstrated the assimilation of in situ observations and focused on the parallelization of the model and analysis scheme for computers with distributed memory. They also showed that regionalization of background covariances has negligible impact on the quality of the analysis.

Several of the most recent publications cited above have proved the feasibility of the ensemble-based methods for real oceanographic problems.

2.2 Other ensemble-based filters

The EnKF can also be related to some other sequential filters such as the Singular Evolutive Extended Kalman (SEEK) filter by Pham et al. (1998), Brasseur et al. (1999), Carmillet et al. (2001) (see also Brusdal et al. 2003, for a comparison of the SEEK and the EnKF); the Reduced Rank Square Root (RRSQRT) filter by Verlaan and Heemink (2001); and the Error Subspace Statistical Estimation (ESSE) by Lermusiaux and Robinsin (1999a,b) and Lermusiaux (2001), which can be interpreted as an EnKF where the analysis is computed in the space spanned by the EOFs of the ensemble.

Anderson (2001) proposed a method denoted the Ensemble Adjustment Kalman Filter, where the analysis is computed without adding perturbations to the observations. If observations are not perturbed in the EnKF this still gives the correct mean of the analyzed ensemble, but results in a too low variance, as explained by Burgers et al. (1998). This is accounted for in the EAKF by deriving a linear operator which replaces the traditional gain matrix and results in an updated ensemble which is consistent with theory. A drawback may be the required inversion of the measurement error covariance when this is nondiagonal. This method becomes a variant of the square root algorithm used by Bishop et al. (2001). It is demonstrated that for small ensembles (10–20 members) the EAKF performs better than the EnKF.

Whitaker and Hamill (2002) proposed another version of the EnKF where the perturbation of observations are avoided. The scheme provides a better estimate of the analysis variance by avoiding the sampling errors of the observation perturbations. The scheme was tested for small ensemble sizes (10–20 members), where it had a clear benefit on the results when compared to the EnKF, which has larger sampling errors with such small ensemble sizes. The scheme is based on a redefinition of the Kalman gain derived from the equation

$$\begin{aligned}\boldsymbol{P}_{\mathrm{e}}^{\mathrm{a}} &= (\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})\boldsymbol{P}^{f}(\boldsymbol{I}-\boldsymbol{H}^{T}\boldsymbol{K}^{T})+\boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{T}\\ &= (\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})\boldsymbol{P}^{f},\end{aligned} \tag{2}$$

where the term $\boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{T}=0$ without perturbations of measurements. A solution of this equation is:

$$\begin{aligned}\boldsymbol{K} = \boldsymbol{P}^{f}\boldsymbol{H}^{T}&\left[\left(\sqrt{\boldsymbol{H}\boldsymbol{P}^{f}\boldsymbol{H}^{T}+\boldsymbol{R}}\right)^{-1}\right]^{T}\\ &\times\left[\sqrt{\boldsymbol{H}\boldsymbol{P}^{f}\boldsymbol{H}^{T}+\boldsymbol{R}}+\sqrt{\boldsymbol{R}}\right]^{-1}.\end{aligned} \tag{3}$$

An explanation of the terms in these equations is given in Section 3. This is essentially a Monte Carlo implementation of the square root filter and was named EnSRF.

A summary of the square root filters by Bishop et al. (2001), Anderson (2001), and Whitaker and Hamill (2002) has been given by Tippet et al. (2003), and see also the general discussion of ensemble methods in a "local least-squares framework" given by Anderson (2003).

2.3 Ensemble smoothers

Some publications have focused on the extension of the EnKF to a smoother formulation. The first formulation was given by van Leeuwen and Evensen (1996), who introduced the Ensemble Smoother (ES). This method was later examined in Evensen (1997) with the Lorenz attractor; applied with a QG model to find a steady mean flow by van Leeuwen (1999b) and for the time-dependent problem in van Leeuwen (2001); and for fish stock assessment by Grønnevik and Evensen (2001). Evensen and van Leeuwen (2000) reexamined the smoother formulation and derived a new algorithm with better properties named the Ensemble Kalman Smoother (EnKS). This method has also been examined in Grønnevik and Evensen (2001) and Brusdal et al. (2003).

2.4 Nonlinear filters and smoothers

Another extension of the EnKF relates to the derivation of an efficient method for solving the nonlinear filtering problem, i.e., taking non-Gaussian contributions in the predicted error statistics into account when computing the analysis. These are discarded in the EnKF (see Evensen and van Leeuwen, 2000), and a fully nonlinear filter is expected to improve the results when used with nonlinear dynamical models with multi-modal behavior where the predicted error statistics are far from Gaussian. Implementations of nonlinear filters based on either kernel approximation or particle interpretations have

been proposed by Miller et al. (1999), Anderson and Anderson (1999), Pham (2001), Miller and Ehret (2002), and van Leeuwen (2003), although more research is needed before these can claimed to be practical for realistic high-dimensional systems.

3 Sequential data assimilation

This section gives a brief introduction to sequential data assimilation methodologies such as the Kalman Filter (KF) and the Extended Kalman Filter (EKF) and outlines the general theory of the EnKF.

3.1 A variance-minimizing analysis

The Kalman Filter is a sequential filter method, which means that the model is integrated forward in time and, whenever measurements are available, these are used to reinitialize the model before the integration continues. We neglect the time index and denote a model forecast and analysis as ψ^f and ψ^a, respectively, and the measurements are contained in $\boldsymbol{d}$. Further, the respective covariances for model forecast, analysis and measurements are denoted $\boldsymbol{P}^f$, $\boldsymbol{P}^a$ and $\boldsymbol{R}$. The analysis equation is then:

$$\psi^a = \psi^f + \boldsymbol{P}^f \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}^f \boldsymbol{H}^T + \boldsymbol{R})^{-1} (\boldsymbol{d} - \boldsymbol{H}\psi^f), \tag{4}$$

with the analysis error covariances given as

$$\boldsymbol{P}^a = \boldsymbol{P}^f - \boldsymbol{P}^f \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}^f \boldsymbol{H}^T + \boldsymbol{R})^{-1} \boldsymbol{H}\boldsymbol{P}^f. \tag{5}$$

Here $\boldsymbol{H}$ is the measurement operator relating the true model state ψ^t to the observations $\boldsymbol{d}$ allowing for measurement errors ϵ, i.e.

$$\boldsymbol{d} = \boldsymbol{H}\psi^t + \epsilon. \tag{6}$$

The reinitialization, ψ^a, is determined as a weighted linear combination of the model prediction, ψ^f, and covariances, $\boldsymbol{P}^f \boldsymbol{H}^T$, corresponding to each of the measurements in $\boldsymbol{d}$. The weights are determined by the error covariance for the model prediction projected onto the measurements, the measurement error covariance, and the difference between the prediction and measurements (i.e., the innovation).

The error covariances for the measurements, $\boldsymbol{R}$, need to be prescribed based on our best knowledge about the accuracy of the measurements and the methodologies used to collect them. The error covariances for the model prediction are computed by solving an equation for the time evolution of the error covariance matrix of the model state.

A derivation of these equations can be found in several publications (see, e.g., Burgers et al. 1998). Note that these equations are often expressed using the so-called Kalman gain matrix

$$\boldsymbol{K} = \boldsymbol{P}^f \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}^f \boldsymbol{H}^T + \boldsymbol{R})^{-1}. \tag{7}$$

3.2 The Kalman Filter

Given a linear dynamical model written on discrete form as:

$$\psi_{k+1} = \boldsymbol{F}\psi_k, \tag{8}$$

the error covariance equation becomes

$$\boldsymbol{P}_{k+1} = \boldsymbol{F}\boldsymbol{P}_k \boldsymbol{F}^T + \boldsymbol{Q}, \tag{9}$$

where the matrix $\boldsymbol{Q}$ is the error covariance matrix for the model errors. The model is assumed to contain errors, e.g., due to neglected physics and numerical approximations. The Eqs. (8) and (9) are integrated to produce the forecasts ψ^f and $\boldsymbol{P}^f$, used in the analysis Eqs. (4) and (5).

3.3 The Extended Kalman Filter

With a nonlinear model

$$\psi_{k+1} = \boldsymbol{f}(\psi_k), \tag{10}$$

the error covariance equation is still Eq. (9) but with $\boldsymbol{F}$ being the tangent linear operator (Jacobian) of $\boldsymbol{f}(\psi)$. Thus, in the Extended Kalman Filter (EKF), a linearized and approximate equation is used for the prediction of error statistics.

3.4 The Ensemble Kalman Filter

The ensemble Kalman filter as proposed by Evensen (1994b) and later clarified by Burgers et al. (1998) is now introduced. We will adapt a three-stage presentation starting with the representation of error statistics using an ensemble of model states, then an alternative to the traditional error covariance equation is proposed for the prediction of error statistics, and finally a consistent analysis scheme is presented.

3.4.1 Representation of error statistics

The error covariance matrices for the forecast and the analyzed estimate, $\boldsymbol{P}^f$ and $\boldsymbol{P}^a$, are defined in the Kalman filter in terms of the true state as

$$\boldsymbol{P}^f = \overline{(\psi^f - \psi^t)(\psi^f - \psi^t)^T}, \tag{11}$$

$$\boldsymbol{P}^a = \overline{(\psi^a - \psi^t)(\psi^a - \psi^t)^T}, \tag{12}$$

where the overline denotes an expectation value, ψ is the model state vector at a particular time and the superscripts f, a, and t represent forecast, analyzed, and true state, respectively. However, the true state is not known, and we therefore define the ensemble covariance matrices around the ensemble mean, $\bar{\psi}$,

$$\boldsymbol{P}^f \simeq \boldsymbol{P}^f_e = \overline{(\psi^f - \overline{\psi}^f)(\psi^f - \overline{\psi}^f)^T}, \tag{13}$$

$$\boldsymbol{P}^a \simeq \boldsymbol{P}^a_e = \overline{(\psi^a - \overline{\psi}^a)(\psi^a - \overline{\psi}^a)^T}, \tag{14}$$

where now the overlines denote an average over the ensemble. Thus, we can use an interpretation where the ensemble mean is the best estimate and the spreading of the ensemble around the mean is a natural definition of the error in the ensemble mean.

Since the error covariances as defined in Eqs. (13) and (14) are defined as ensemble averages, there will clearly exist infinitively many ensembles with an error covariance equal to $\boldsymbol{P}_e^f$ and $\boldsymbol{P}_e^a$. Thus, instead of storing a full covariance matrix, we can represent the same error statistics using an appropriate ensemble of model states. Given an error covariance matrix, an ensemble of finite size will always provide an approximation to the error covariance matrix. However, when the size of the ensemble N increases, the errors in the Monte Carlo sampling will decrease proportional to $1/\sqrt{N}$.

Suppose now that we have N model states in the ensemble, each of dimension n. Each of these model states can be represented as a single point in an n-dimensional state space. All the ensemble members together will constitute a cloud of points in the state space. Such a cloud of points in the state space can, in the limit when N goes to infinity, be described using a probability density function

$$\phi(\psi) = \frac{dN}{N}, \tag{15}$$

where dN is the number of points in a small unit volume and N is the total number of points. With knowledge about either ϕ or the ensemble representing ϕ, we can calculate whichever statistical moments (such as mean, covariances, etc.) we want whenever they are needed.

The conclusion so far is that the information contained by a full probability density function can be exactly represented by an infinite ensemble of model states.

3.4.2 Prediction of error statistics

The EnKF was designed to resolve two major problems related to the use of the EKF with nonlinear dynamics in large state spaces. The first problem relates to the use of an approximate closure scheme in the EKF, and the other to the huge computational requirements associated with the storage and forward integration of the error covariance matrix.

The EKF applies a closure scheme where third- and higher-order moments in the error covariance equation are discarded. This linearization has been shown to be invalid in a number of applications, e.g., Evensen (1992) and Miller et al. (1994). In fact, the equation is no longer the fundamental equation for the error evolution when the dynamical model is nonlinear. In Evensen (1994b) it was shown that a Monte Carlo method can be used to solve an equation for the time evolution of the probability density of the model state, as an alternative to using the approximate error covariance equation in the EKF.

For a nonlinear model where we appreciate that the model is not perfect and contains model errors, we can write it as a stochastic differential equation (on continuous form) as:

$$d\psi = \boldsymbol{f}(\psi)dt + \boldsymbol{g}(\psi)d\boldsymbol{q}. \tag{16}$$

This equation states that an increment in time will yield an increment in ψ, which in addition, is influenced by a random contribution from the stochastic forcing term, $\boldsymbol{g}(\psi)d\boldsymbol{q}$, representing the model errors. The $d\boldsymbol{q}$ describe a vector Brownian motion process with covariance $\boldsymbol{Q}dt$. Because the model is nonlinear, $\boldsymbol{g}$ is not an explicit function of the random variable $d\boldsymbol{q}$, so the Ito interpretation of the stochastic differential equation has to be used instead of the Stratonovich interpretation (see Jazwinski 1970)

When additive Gaussian model errors forming a Markov process are used, one can derive the Fokker–Planck equation (also named Kolmogorov's equation) which describes the time evolution of the probability density $\phi(\psi)$ of the model state,

$$\frac{\partial\phi}{\partial t} + \sum_i \frac{\partial(f_i\phi)}{\partial\psi_i} = \frac{1}{2}\sum_{i,j}\frac{\partial^2\phi(\boldsymbol{g}\boldsymbol{Q}\boldsymbol{g}^T)_{ij}}{\partial\psi_i\partial\psi_j}, \tag{17}$$

where f_i is the component number i of the model operator $\boldsymbol{f}$ and $\boldsymbol{g}\boldsymbol{Q}\boldsymbol{g}^T$ is the covariance matrix for the model errors.

This equation does not apply any important approximations and can be considered as the fundamental equation for the time evolution of error statistics. A detailed derivation is given in Jazwinsky (1970). The equation describes the change of the probability density in a local "volume" which is dependent on the divergence term describing a probability flux into the local "volume" (impact of the dynamical equation) and the diffusion term which tends to flatten the probability density due to the effect of stochastic model errors. If Eq. (17) could be solved for the probability density function, it would be possible to calculate statistical moments like the mean state and the error covariance for the model forecast to be used in the analysis scheme.

The EnKF applies a so-called Markov Chain Monte Carlo (MCMC) method to solve Eq (17). The probability density can be represented using a large ensemble of model states. By integrating these model states forward in time according to the model dynamics described by the stochastic differential Eq. (16), this ensemble prediction is equivalent to solving the Fokker–Planck equation using an MCMC method. This procedure forms the backbone for the EnKF.

A linear model for a Gauss–Markov process in which the initial condition is assumed to be taken from a normal distribution will have a probability density which is completely characterized by its mean and covariance for all times. One can then derive exact equations for the evolution of the mean and the covariance as a simpler alternative than solving the full Kolmogorov's equation. Such moments of Kolmogorov's

equation, including the error covariance Eq. (9), are easy to derive, and several methods are illustrated by Jazwinski (1970, examples 4.19–4.21). This is actually what is done in the KF and EKF.

For a nonlinear model, the mean and covariance matrix will not in general characterize $\phi(\psi, t)$. They do, however, determine the mean path and the dispersion about that path, and it is possible to solve approximate equations for the moments, which is the procedure characterizing the extended Kalman filter.

An alternative to the approximate stochastic dynamic approach for solving Kolmogorov's equation and predicting the error statistics is to use Monte Carlo methods. A large cloud of model states (points in state space) can be used to represent a specific probability density function. By integrating such an ensemble of states forward in time, it is easy to calculate approximate estimates for moments of the probability density function at different time levels. In this context the Monte Carlo method might be considered a particle method in the state space.

3.4.3 An analysis scheme

The KF analysis scheme is using the definitions of $\boldsymbol{P}^{\mathrm{f}}$ and $\boldsymbol{P}^{\mathrm{a}}$ as given by Eqs. (11) and (12). We will now give a derivation of the analysis scheme where the ensemble covariances are used as defined by Eqs. (13) and (14). This is convenient, since in practical implementations one is doing exactly this, and it will also lead to a consistent formulation of the EnKF.

As will be shown later, it is essential that the observations are treated as random variables having a distribution with mean equal to the first-guess observations and covariance equal to $\boldsymbol{R}$. Thus, we start by defining an ensemble of observations

$$\boldsymbol{d}_j = \boldsymbol{d} + \epsilon_j, \tag{18}$$

where j counts from 1 to the number of model state ensemble members N. It is ensured that the simulated random measurement errors have mean equal to zero. Next, we define the ensemble covariance matrix of the measurements as

$$\boldsymbol{R}_e = \overline{\epsilon\epsilon^T}, \tag{19}$$

and, of course, in the limit of an infinite ensemble this matrix will converge toward the prescribed error covariance matrix $\boldsymbol{R}$ used in the standard Kalman filter.

The following discussion is valid both using an exactly prescribed $\boldsymbol{R}$ and an ensemble representation $\boldsymbol{R}_e$ of $\boldsymbol{R}$. The use of $\boldsymbol{R}_e$ introduces an additional approximation which becomes convenient when implementing the analysis scheme. This can be justified by the fact that the actual observation error covariance matrix is poorly known and the errors introduced by the ensemble representation can be made less than the initial uncertainty in the exact form of $\boldsymbol{R}$ by choosing a large enough ensemble size. Further, the errors introduced by using an ensemble representation for $\boldsymbol{R}$ have less impact than the use of an ensemble representation for $\boldsymbol{P}$. $\boldsymbol{R}$ only appears in the computation of the coefficients for the influence functions $\boldsymbol{P}^f\boldsymbol{H}^T$ while $\boldsymbol{P}$ both appears in the computation of the coefficients and determines the influence functions.

The analysis step for the EnKF consists of the following updates performed on each of the model state ensemble members

$$\psi_j^a = \psi_j^f + \boldsymbol{P}_e^f\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_e^f\boldsymbol{H}^T + \boldsymbol{R}_e)^{-1}(\boldsymbol{d}_j - \boldsymbol{H}\psi_j^f). \tag{20}$$

With a finite ensemble size, this equation will be an approximation. Further, if the number of measurements is larger than the number of ensemble members, the matrices $\boldsymbol{H}\boldsymbol{P}_e^f\boldsymbol{H}^T$ and $\boldsymbol{R}_e$ will be singular, and a pseudo inversion must be used. Note that Eq. (20) implies that

$$\overline{\psi^a} = \overline{\psi^f} + \boldsymbol{P}_e^f\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_e^f\boldsymbol{H}^T + \boldsymbol{R}_e)^{-1}(\overline{\boldsymbol{d}} - \boldsymbol{H}\overline{\psi^f}), \tag{21}$$

where $\overline{\boldsymbol{d}} = \boldsymbol{d}$ is the first guess vector of measurements. Thus, the relation between the analyzed and forecast ensemble mean is identical to the relation between the analyzed and forecasted state in the standard Kalman filter in Eq. (4), apart from the use of $\boldsymbol{P}_e^{f,a}$ and $\boldsymbol{R}_e$ instead of $\boldsymbol{P}^{f,a}$ and $\boldsymbol{R}$. Note that the introduction of an ensemble of observations does not make any difference for the update of the ensemble mean, since this does not affect Eq. (21).

If the mean, $\overline{\psi^a}$, is considered to be the best estimate, then it is an arbitrary choice whether one updates the mean using the first-guess observations $\boldsymbol{d}$, or if one updates each of the ensemble members using the perturbed observations (Eq. 18). However, it will now be shown that by updating each of the ensemble members using the perturbed observations, one also creates a new ensemble having the correct error statistics for the analysis. The updated ensemble can then be integrated forward in time till the next observation time.

Moreover, the error covariance, $\boldsymbol{P}_e^a$, of the analyzed ensemble is reduced in the same way as in the standard Kalman Filter. We now derive the analyzed error covariance estimate resulting from the analysis scheme given above, but using the standard Kalman filter form for the analysis equations. First, note that Eqs. (20) and (21) are used to obtain:

$$\psi_j^a - \overline{\psi^a} = (\boldsymbol{I} - \boldsymbol{K}_e\boldsymbol{H})(\psi_j^f - \overline{\psi^f}) + \boldsymbol{K}_e(\boldsymbol{d}_j - \overline{\boldsymbol{d}}), \tag{22}$$

where we have used the definition of the Kalman gain,

$$\boldsymbol{K}_e = \boldsymbol{P}_e^f\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_e^f\boldsymbol{H}^T + \boldsymbol{R}_e)^{-1}. \tag{23}$$

The derivation is then as follows,

$$\begin{aligned}\boldsymbol{P}_e^a &= \overline{(\psi^a - \overline{\psi^a})(\psi^a - \overline{\psi^a})^T} \\ &= (\boldsymbol{I} - \boldsymbol{K}_e\boldsymbol{H})\boldsymbol{P}_e^f(\boldsymbol{I} - \boldsymbol{H}^T\boldsymbol{K}_e^T) + \boldsymbol{K}_e\boldsymbol{R}_e\boldsymbol{K}_e^T \\ &= \boldsymbol{P}_e^f - \boldsymbol{K}_e\boldsymbol{H}\boldsymbol{P}_e^f - \boldsymbol{P}_e^f\boldsymbol{H}^T\boldsymbol{K}_e^T + \boldsymbol{K}_e(\boldsymbol{H}\boldsymbol{P}_e^f\boldsymbol{H}^T + \boldsymbol{R}_e)\boldsymbol{K}_e^T \\ &= (\boldsymbol{I} - \boldsymbol{K}_e\boldsymbol{H})\boldsymbol{P}_e^f.\end{aligned} \tag{24}$$

The last expression in this equation is the traditional result for the minimum variance error covariance found in the KF analysis scheme. This implies that the EnKF in the limit of an infinite ensemble size will give exactly the same result in the computation of the analysis as the KF and EKF. Note that this derivation clearly states that the observations $\boldsymbol{d}$ must be treated as random variables to obtain the measurement error covariance matrix $\boldsymbol{R}_e$ into the expression. It has been assumed that the distributions used to generate the model state ensemble and the observation ensemble are independent.

3.4.4 Summary

We now have a complete system of equations which constitutes the ensemble Kalman filter (EnKF), and the resemblance with the standard Kalman filter is maintained. This is also true for the forecast step. Each ensemble member evolves in time according to the model dynamics. The ensemble covariance matrix of the errors in the model equations, given by

$$\boldsymbol{Q}_e = \overline{d\boldsymbol{q}_k d\boldsymbol{q}_k^T}, \tag{25}$$

converges to $\boldsymbol{Q}$ in the limit of infinite ensemble size. The ensemble mean then evolves according to the equation

$$\begin{aligned}\overline{\psi_{k+1}} &= \overline{\boldsymbol{f}(\psi_k)} \\ &= \boldsymbol{f}(\overline{\psi_k}) + \text{n.l.},\end{aligned} \tag{26}$$

where n.l. represents the terms which may arise if $\boldsymbol{f}$ is non-linear. One of the advantages of the EnKF is that the effect of these terms is retained, since each ensemble member is integrated independently by the model.

The error covariance of the ensemble evolves according to

$$\boldsymbol{P}_e^{k+1} = \boldsymbol{F}\boldsymbol{P}_e^k\boldsymbol{F}^T + \boldsymbol{Q}_e + \text{n.l.}, \tag{27}$$

where $\boldsymbol{F}$ is the tangent linear operator evaluated at the current time step. This is again an equation of the same form as is used in the standard Kalman filter, except of the extra terms n.l. that may appear if $\boldsymbol{f}$ is nonlinear. Implicitly, the EnKF retains these terms also for the error covariance evolution.

For a linear dynamical model, the sampled $\boldsymbol{P}_e$ converges to $\boldsymbol{P}$ for infinite ensemble sizes and, independent from the model, $\boldsymbol{R}_e$ converges to $\boldsymbol{R}$ and $\boldsymbol{Q}_e$ converges to $\boldsymbol{Q}$. Thus, in this limit, both algorithms, the KF and the EnKF, are equivalent.

For nonlinear dynamics the so-called extended Kalman filter may be used and is given by the evolution Eqs. (26) and (27) with the n.l. terms neglected. Ensemble-based filters include the full effect of these terms and there are no linearizations or closure assumptions applied. In addition, there is no need for a tangent linear operator, such as $\boldsymbol{F}$, or its adjoint, and this makes these methods very easy to implement for practical applications.

This leads to an interpretation of the EnKF as a purely statistical Monte Carlo method where the ensemble of model states evolves in state space with the mean as the best estimate and the spreading of the ensemble as the error variance. At measurement times each observation is represented by another ensemble, where the mean is the actual measurement and the variance of the ensemble represents the measurement errors.

4 Practical formulation and interpretation

This section discusses the EnKF in more detail with focus on the practical formulation and interpretation. It is shown that an interpretation in the "ensemble space" provides a better understanding of the actual algorithm and also allows for very efficient algorithms to be developed.

4.1 The initial ensemble

The initial ensemble should ideally be chosen to properly represent the error statistics of the initial guess for the model state. However, a modest mis-specification of the initial ensemble normally does not influence the results very much over time. The rule of thumb seems to be that one needs to create an ensemble of model states by adding some kind of perturbations to a best-guess estimate, and then integrate the ensemble over a time interval covering a few characteristic time scales of the dynamical system. This will ensure that the system is in dynamical balance and that proper multivariate correlations have developed.

The perturbations can be created in different ways. The simplest is to sample random numbers (for a scalar model), random curves (for a 1-D model) or random fields (for a model with 2 or higher dimensions), from a specified distribution. In Appendix E there is an example of a procedure for generating such random perturbations.

4.2 The ensemble integration

The ensemble of model states is integrated forward in time according to the stochastic Eq. (16). In a practical implementation this becomes just a standard integration of the numerical model but subject to a stochastic noise which resembles the uncertainties in the model. Note that the EnKF allows for a wide range of noise models. Stochastic terms can be added to all poorly known model parameters and one is not restricted to using Gaussian distributed noise. Further, it is possible to use time-correlated (red) noise by transforming it into white noise, as is explained in the following section. A different noise model will change the form of the stochastic

Eq. (16) and also lead to a different form of the Fokker–Planck Eq. (17). However, the Fokker–Planck equation is never used explicitly in the algorithm and the EnKF would still provide a Monte Carlo method for solving it.

4.2.1 Simulation of model errors

The following equation can be used for simulating the time evolution of model errors:

$$\boldsymbol{q}_k = \alpha \boldsymbol{q}_{k-1} + \sqrt{1-\alpha^2}\boldsymbol{w}_{k-1}. \tag{28}$$

Here we assume that $\boldsymbol{w}_k$ is a sequence of white noise drawn from a distribution of smooth pseudorandom fields with mean equal to 0 and variance equal to 1. Such fields can be generated using the algorithm presented in Appendix E. The coefficient $\alpha \in [0,1)$ determines the time decorrelation of the stochastic forcing, e.g., $\alpha = 0$ generates a sequence which is white in time, while $\alpha = 1$ will remove the stochastic forcing and represent the model errors with a random field which is constant in time.

This equation ensures that the variance of the ensemble of $\boldsymbol{q}_k$s is equal to 1 as long as the variance of the ensemble of $\boldsymbol{q}_{k-1}$s is 1. Thus, this equation will produce a sequence of time-correlated pseudorandom fields with mean equal to zero and variance equal to 1.

The covariance in time between $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$, determined by Eq. (28), is

$$\overline{\boldsymbol{q}_i \boldsymbol{q}_j} = \alpha^{|i-j|}. \tag{29}$$

Determination of α. The factor α should be related to the time step used and a specified time decorrelation length τ. The Eq. (28), when excluding the stochastic term, resembles a difference approximation to

$$\frac{\partial q}{\partial t} = -\frac{1}{\tau} q, \tag{30}$$

which states that q is damped with a ratio e^{-1} over a time period $t = \tau$. A numerical approximation becomes

$$q_k = \left(1 - \frac{\Delta t}{\tau}\right) q_{k-1}, \tag{31}$$

where Δt is the time step. Thus, we define α as

$$\alpha = 1 - \frac{\Delta t}{\tau}, \tag{32}$$

where $\tau \geq \Delta t$.

Physical model. Based on random walk theory (see below), the physical model can be written as

$$\psi_k = \boldsymbol{f}(\psi_{k-1}) + \sqrt{\Delta t}\sigma\rho\boldsymbol{q}_k, \tag{33}$$

where σ is the standard deviation of the model error and ρ is a factor to be determined. The choice of the stochastic term is explained next.

Variance growth due to the stochastic forcing. To explain the choice of the stochastic term in Eq. (33) we will use a simple random walk model for illustration, i.e.,

$$\psi_k = \psi_{k-1} + \sqrt{\Delta t}\sigma\rho\boldsymbol{q}_k. \tag{34}$$

This equation can be rewritten as:

$$\psi_k = \psi_0 + \sqrt{\Delta t}\sigma\rho \sum_{i=0}^{k-1} \boldsymbol{q}_{i+1}. \tag{35}$$

The variance can be found by squaring (35) and taking the ensemble average, i.e.,

$$\overline{\psi_n \psi_n^T} = \overline{\psi_0 \psi_0^T} + \Delta t \sigma^2 \rho^2 \overline{\left(\sum_{k=0}^{n-1} \boldsymbol{q}_{k+1}\right)\left(\sum_{k=0}^{n-1} \boldsymbol{q}_{k+1}\right)^T} \tag{36}$$

$$= \overline{\psi_0 \psi_0^T} + \Delta t \sigma^2 \rho^2 \sum_{j=0}^{n-1}\sum_{i=0}^{n-1} \overline{\boldsymbol{q}_{i+1}\boldsymbol{q}_{j+1}^T} \tag{37}$$

$$= \overline{\psi_0 \psi_0^T} + \Delta t \sigma^2 \rho^2 \sum_{j=0}^{n-1}\sum_{i=0}^{n-1} \alpha^{|i-j|} \tag{38}$$

$$= \overline{\psi_0 \psi_0^T} + \Delta t \sigma^2 \rho^2 \left(-n + 2\sum_{i=0}^{n-1}(n-i)\alpha^i\right) \tag{39}$$

$$= \overline{\psi_0 \psi_0^T} + \Delta t \sigma^2 \rho^2 \frac{n - 2\alpha - n\alpha^2 + 2\alpha^{n+1}}{(1-\alpha)^2}, \tag{40}$$

where the expression (29) has been used. Note that n here denotes the number of time steps and not the dimension of the model state as in the remainder of this paper. The double sum in Eq. (38) is just summing elements in a matrix and is replaced by a single sum operating on diagonals of constant values. The summation in Eq. (39) has an explicit solution (Gradshteyn and Ryzhik 1979, formula 0.113).

If the sequence of model noise $\boldsymbol{q}_k$ is white in time ($\alpha = 0$), this equation implies an increase in variance equal to $\sigma^2\rho^2$ when Eq. (34) is iterated n time steps of length Δt, over one time unit ($n\Delta t = 1$). Thus, in this case $\rho = 1$ is a natural choice, since this leads to the correct increase in ensemble variance given by σ^2.

In the case with red model errors, the increase in ensemble variance over one time unit will increase up to a maximum of $\sigma^2\rho^2/\Delta t$ in the case when $\alpha = 1$ (not covered by Eq. 39).

The two Eqs. (28) and (33) provide the standard framework for introducing stochastic model errors when using the EnKF. The formula (40) provides the mean for scaling the perturbations in Eq. (33) when changing α and/or the number of time steps per time unit, n, to ensure that the ensemble variance growth over a time unit remains the same.

Thus, the constraint that

$$1 = \rho^2 \Delta t \frac{n - 2\alpha - n\alpha^2 + 2\alpha^{n+1}}{(1-\alpha)^2}, \tag{41}$$

defines the factor

$$\rho^2 = \frac{1}{\Delta t}\frac{(1-\alpha)^2}{n - 2\alpha - n\alpha^2 + 2\alpha^{n+1}}, \tag{42}$$

which ensures that the variance growth over time becomes independent of α and Δt (as long as the dynamical model is linear).

4.2.2 Estimation of model errors

When red model noise is used, correlations will develop between the red noise and the model variables. Thus, during the analysis it is also possible to consistently update the model noise as well as the model state. This was illustrated in an example by Reichle et al. (2002). We introduce a new state vector which consists of ψ augmented with $\boldsymbol{q}$. The two Eqs. (28) and (33) can then be written as

$$\begin{pmatrix} \boldsymbol{q}_k \\ \psi_k \end{pmatrix} = \begin{pmatrix} \alpha \boldsymbol{q}_{k-1} \\ \boldsymbol{f}(\psi_{k-1}) + \sqrt{\Delta t}\sigma\rho\boldsymbol{q}_k \end{pmatrix} + \begin{pmatrix} \sqrt{1-\alpha^2}\boldsymbol{w}_{k-1} \\ 0 \end{pmatrix}. \tag{43}$$

During the analysis we can now compute covariances between the observed model variable and the model noise vector $\boldsymbol{q}$ and update this together with the state vector. This will lead to a correction of the mean of $\boldsymbol{q}$ as well as a reduction of the variance in the model noise ensemble. Note that this procedure estimates the actual error in the model for each ensemble member, given the prescribed model error statistics.

The form of Eq. (28) ensures that, over time, $\boldsymbol{q}_k$ will approach a distribution with mean equal to zero and variance equal to one, as long as we do not update $\boldsymbol{q}_k$ in the analysis scheme.

For an illustration of the use of time correlated model errors and their estimation we refer to Appendix F.

4.3 The EnKF analysis scheme

This section attempts to explain in some detail how the EnKF analysis can be computed efficiently for practical applications. In particular, it discusses how the filter can be used to compute a global analysis at an affordable cost, even with a very large number of measurements. It presents a storage scheme which requires only one copy of the ensemble to be kept in memory, and an efficient algorithm for computation of the expensive final matrix multiplication. The concept of a local analysis is discussed in Section 4.4. A discussion is also given on the assimilation of nonlinear measurements in Section 4.5, a problem which is solved by augmenting the model state with the model's measurement equivalents. Moreover, this algorithm also allows for the efficient assimilation of in situ measurements in a consistent manner where one entirely relies on the ensemble predicted error statistics (see Appendix C). Finally a discussion is given on the assimilation of nonsynoptic measurements in Section 4.6.

4.3.1 Definitions and the analysis equation

Define the matrix holding the ensemble members $\psi_i \in \Re^n$,

$$\boldsymbol{A} = (\psi_1, \psi_2, \ldots, \psi_N) \in \Re^{n\times N}, \tag{44}$$

where N is the number of ensemble members and n is the size of the model state vector.

The ensemble mean is stored in each column of $\overline{\boldsymbol{A}}$ which can be defined as

$$\overline{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{1}_N, \tag{45}$$

where $\boldsymbol{1}_N \in \Re^{N\times N}$ is the matrix where each element is equal to $1/N$. We can then define the ensemble perturbation matrix as

$$\boldsymbol{A}' = \boldsymbol{A} - \overline{\boldsymbol{A}} = \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{1}_N). \tag{46}$$

The ensemble covariance matrix $\boldsymbol{P}_e \in \Re^{n\times n}$ can be defined as

$$\boldsymbol{P}_e = \frac{\boldsymbol{A}'(\boldsymbol{A}')^T}{N-1}. \tag{47}$$

Given a vector of measurements $\boldsymbol{d} \in \Re^m$, with m being the number of measurements, we can define the N vectors of perturbed observations as

$$\boldsymbol{d}_j = \boldsymbol{d} + \epsilon_j, \quad j = 1, \ldots, N, \tag{48}$$

which can be stored in the columns of a matrix

$$\boldsymbol{D} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N) \in \Re^{m\times N}, \tag{49}$$

while the ensemble of perturbations, with ensemble mean equal to zero, can be stored in the matrix

$$\Upsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_N) \in \Re^{m\times N}, \tag{50}$$

from which we can construct the ensemble representation of the measurement error covariance matrix

$$\boldsymbol{R}_e = \frac{\Upsilon\Upsilon^T}{N-1}. \tag{51}$$

The standard analysis equation, expressed in terms of the ensemble covariance matrices, is

$$\boldsymbol{A}^a = \boldsymbol{A} + \boldsymbol{P}_e\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_e\boldsymbol{H}^T + \boldsymbol{R}_e)^{-1}(\boldsymbol{D} - \boldsymbol{H}\boldsymbol{A}). \tag{52}$$

Using the ensemble of innovation vectors defined as:

$$\boldsymbol{D}' = \boldsymbol{D} - \boldsymbol{H}\boldsymbol{A} \tag{53}$$

and the definitions of the ensemble error covariance matrices in Eqs. (51) and (47) the analysis can be expressed as:

$$\boldsymbol{A}^a = \boldsymbol{A} + \boldsymbol{A}'\boldsymbol{A}'^T\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{A}'\boldsymbol{A}'^T\boldsymbol{H}^T + \Upsilon\Upsilon^T\right)^{-1}\boldsymbol{D}'. \tag{54}$$

The potential singularity of the inverse computation requires the use of a pseudo inverse and the practical implementation is discussed next.

4.3.2 Practical formulation and implementation

The traditional way of solving the analysis Eq. (54) would involve the computation of the eigenvalue decomposition directly from the $m \times m$ matrix,

$$\boldsymbol{H}\boldsymbol{A}'\boldsymbol{A}'^{T}\boldsymbol{H}^{T} + \Upsilon\Upsilon^{T} = \boldsymbol{Z}\Lambda\boldsymbol{Z}^{T}, \tag{55}$$

which has the inverse (or pseudo inverse if the matrix is singular)

$$(\boldsymbol{H}\boldsymbol{A}'\boldsymbol{A}'^{T}\boldsymbol{H}^{T} + \Upsilon\Upsilon^{T})^{-1} = \boldsymbol{Z}\Lambda^{-1}\boldsymbol{Z}^{T}. \tag{56}$$

The cost of the eigenvalue decomposition is proportional to m^2 and becomes unaffordable for large m. Note, however, that the rank of $\boldsymbol{Z}\Lambda\boldsymbol{Z}^{T}$ is less than or equal to N. Thus, Λ will have N or less non-zero eigenvalues and it may therefore be possible to use a more efficient eigenvalue decomposition algorithm which computes and stores only the first N columns of $\boldsymbol{Z}$.

It is important to note that if different measurement types are assimilated simultaneously, the observed model variables need to be made nondimensional or scaled to have similar variability. This is required to ensure that the eigenvalues corresponding to each of the measurement types have the same magnitude. The standard approach for resolving this is to assimilate different measurement types, which normally have uncorrelated errors, sequentially one dataset at a time. The validity of this approach has been shown, e.g., by Evensen and van Leeuwen (2000). This ensures that the results are not affected by a poor scaling, which in the worst case may result in the truncation of all eigenvalues corresponding to measurements of one kind.

Alternative solution for large m. If the perturbations used for measurements are chosen such that

$$\boldsymbol{H}\boldsymbol{A}'\Upsilon^{T} \equiv 0, \tag{57}$$

meaning that the ensemble perturbations and the measurement errors are uncorrelated (equivalent to the common assumption of uncorrelated forecast and measurement errors), then the following is valid

$$\boldsymbol{H}\boldsymbol{A}'\boldsymbol{A}'^{T}\boldsymbol{H}^{T} + \Upsilon\Upsilon^{T} = (\boldsymbol{H}\boldsymbol{A}' + \Upsilon)(\boldsymbol{H}\boldsymbol{A}' + \Upsilon)^{T}. \tag{58}$$

This is an important point since it means that the inverse can be computed to a cost proportional to mN rather than m^2. This is seen by the following: first compute the singular value decomposition (SVD) of the $m \times N$ matrix

$$\boldsymbol{H}\boldsymbol{A}' + \Upsilon = \boldsymbol{U}\Sigma\boldsymbol{V}^{T}. \tag{59}$$

The Eq. (58) then becomes

$$\boldsymbol{H}\boldsymbol{A}'\boldsymbol{A}'^{T}\boldsymbol{H}^{T} + \Upsilon\Upsilon^{T} = \boldsymbol{U}\Sigma\boldsymbol{V}^{T}\boldsymbol{V}\Sigma^{T}\boldsymbol{U}^{T} = \boldsymbol{U}\Sigma\Sigma^{T}\boldsymbol{U}^{T}. \tag{60}$$

Here the product $\Sigma\Sigma^{T}$ will be identical to the upper left $N \times N$ quadrant of Λ, which corresponds to the N nonzero eigenvalues. Further, the N singular vectors contained in $\boldsymbol{U}$ are also identical to the N first eigenvectors in $\boldsymbol{Z}$. Thus, the inverse is again Eq. (56). The numerical cost is now proportional to mN, which is a huge benefit when m is large. This procedure allows us to efficiently compute the inversion for a global analysis in most practical situations.

Update costs. As soon as the inversion just discussed has been completed, the analysis can be computed from

$$\boldsymbol{A}^{a} = \boldsymbol{A} + \boldsymbol{A}'(\boldsymbol{H}\boldsymbol{A}')^{T}\boldsymbol{U}\Lambda^{-1}\boldsymbol{U}^{T}\boldsymbol{D}'. \tag{61}$$

The matrix Λ^{-1} will have nonzero elements only on the diagonal. If we use the pseudo inverse taking into account e.g., 99% of the variance, only the first few $p \leq N$, terms will be nonzero since the rank of the inverted matrix is $p \leq N$ from Eq. (58). This can be exploited using the following scheme:

$$\boldsymbol{X}_1 = \Lambda^{-1}\boldsymbol{U}^{T} \quad \in \Re^{N\times m} \quad mp, \tag{62}$$

$$\boldsymbol{X}_2 = \boldsymbol{X}_1\boldsymbol{D}' \quad \in \Re^{N\times N} \quad mNp, \tag{63}$$

$$\boldsymbol{X}_3 = \boldsymbol{U}\boldsymbol{X}_2 \quad \in \Re^{m\times N} \quad mNp, \tag{64}$$

$$\boldsymbol{X}_4 = (\boldsymbol{H}\boldsymbol{A}')^{T}\boldsymbol{X}_3 \quad \in \Re^{N\times N} \quad mNN, \tag{65}$$

$$\boldsymbol{A}^{a} = \boldsymbol{A} + \boldsymbol{A}'\boldsymbol{X}_4 \quad \in \Re^{n\times N} \quad nNN, \tag{66}$$

where the last two columns denote the dimension of the resulting matrix and the number of floating point operations needed to compute it. Since $p \leq N$ and $m \ll n$ for all practical applications, the dominant cost is now the last computation, which is nN^2, and which is independent of m. All the steps including the singular value decomposition have a cost which is linear in the number of measurements rather than quadratic. A practical approach for performing this last multiplication will be discussed later.

If we use a full rank matrix, $\boldsymbol{H}\boldsymbol{P}_e\boldsymbol{H}^{T} + \boldsymbol{R}$, where $\boldsymbol{R}$ is not represented using an ensemble of perturbations, the computation of the analysis will be significantly more expensive. First, the full matrix $\boldsymbol{H}\boldsymbol{P}_e\boldsymbol{H}^{T} = (\boldsymbol{H}\boldsymbol{A}')(\boldsymbol{H}\boldsymbol{A}')^{T}$ must be constructed to a cost of m^2N, followed by the eigenvalue decomposition Eq. (55), which requires another $\mathcal{O}(m^2)$ floating point operations. In this case, the steps (63) and (64) also come at a cost of m^2N. Thus, the introduction of low rank by representing the measurement error covariance matrix with an ensemble of perturbations leads to a significant saving by transforming all the m^2N operations to be linear in m.

4.3.3 The case with $m \ll N$.

The algorithm described above is optimized for the case when $m \gg N$. In the case when $m \ll N$, a small modification is appropriate. First, note that even if the eigenvalue factorization in Eq. (55) now becomes less expensive than the singular value decomposition in Eq. (59), the construction of the full matrix $\boldsymbol{H}\boldsymbol{A}'(\boldsymbol{H}\boldsymbol{A}')^{T}$ is even more expensive (requires m^2N floating point operations). Thus, it is still beneficial to use the ensemble representation, Υ, for the measurement error statistics and to compute the SVD using the algorithm described above.

It was shown above that the dominant cost is associated with the final computation in Eq. (66) which is nN^2. This can now be reduced by a reordering of the multiplications in Eq. (61). After $\boldsymbol{X}_3$ is computed, the equation is

$$\boldsymbol{A}^a = \boldsymbol{A} + \boldsymbol{A}'(\boldsymbol{H}\boldsymbol{A}')^T\boldsymbol{X}_3, \tag{67}$$

where the matrix dimensions in the last term can be written as $(n \times N)(N \times m)(m \times N)$. Computing the multiplications from left to right requires $2nmN$ operations, while computing them from right to left requires $(m+n)N^2$ operations. Thus, for a small number of measurements when $2nmN < (m+n)N^2$ it is more efficient to compute the influence functions $\boldsymbol{A}'(\boldsymbol{H}\boldsymbol{A}')^T = \boldsymbol{P}_e\boldsymbol{H}^T$ first, and then add these to the forecast ensemble using the coefficients contained in $\boldsymbol{X}_3$.

4.3.4 Remarks on analysis equation

The Eq. (66) expresses the analysis as a first guess plus a combination of ensemble perturbations, i.e., $\boldsymbol{A}'\boldsymbol{X}_4$. From the discussion above we could also write the analysis equation as:

$$\boldsymbol{A}^a = \boldsymbol{A} + \boldsymbol{A}'(\boldsymbol{H}\boldsymbol{A}')^T\boldsymbol{X}_3 = \boldsymbol{A} + \boldsymbol{P}_e\boldsymbol{H}^T(N-1)\boldsymbol{X}_3. \tag{68}$$

This is the standard notation used in Kalman filters where one measures the error covariance matrix to compute the influence functions, one for each measurement, which are added to the forecast.

Note also that the Eq. (66) can be written as

$$\boldsymbol{A}^a = \boldsymbol{A} + (\boldsymbol{A} - \overline{\boldsymbol{A}})\boldsymbol{X}_4 \tag{69}$$

$$= \boldsymbol{A} + \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{1}_N)\boldsymbol{X}_4 \tag{70}$$

$$= \boldsymbol{A}(\boldsymbol{I} + \boldsymbol{X}_4) \tag{71}$$

$$= \boldsymbol{A}\boldsymbol{X}_5, \tag{72}$$

where we have used $\boldsymbol{1}_N\boldsymbol{X}_4 \equiv \boldsymbol{0}$. Obviously, the first observation to make is that the analyzed ensemble becomes a weakly nonlinear combination of the predicted ensemble. The notation "weakly nonlinear combination" is used since the $\boldsymbol{X}_5$ matrix depends on the forecast ensemble only through the projection of $\boldsymbol{A}$ onto the measurements, i.e., $\boldsymbol{A}\boldsymbol{H}^T$. It then becomes of interest to examine $\boldsymbol{X}_5$ to study the properties of this particular combination. Each column of $\boldsymbol{X}_5$ will hold the coefficients defining the corresponding new ensemble member. For this estimate to be unbiased, the sum of each column of $\boldsymbol{X}_5$ should be equal to 1, which is actually a good test for the numerical coding leading to $\boldsymbol{X}_5$. Also, one can in most applications expect that $\boldsymbol{X}_5$ is diagonal dominant since the diagonal holds the coefficient for the first-guess ensemble member, while all off-diagonal elements introduce corrections imposed by the measurements. By examining the rows of the matrix $\boldsymbol{X}_5$, one can determine if some ensemble members appear to be more important than others. Note that the off-diagonal elements in $\boldsymbol{X}_5$ will also have negative values.

Computation of the mean of the analyzed ensemble can be written as follows:

$$\overline{\psi^a} = \frac{1}{N}\sum_{j=1}^{N}\psi_j^a, \tag{73}$$

$$= \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\psi_i X_{ij}, \tag{74}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\psi_i\sum_{j=1}^{N}X_{ij}, \tag{75}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\psi_i y_i, \quad \text{where } y_i = \sum_{j=1}^{N}X_{ij}. \tag{76}$$

Thus, the sum, y_i, of the elements in each row in $\boldsymbol{X}_5$ defines the coefficients for the combination of forecast members defining the mean of the analysis. The y_i values therefore also determine which of the ensemble members contributes most strongly to the analysis.

If we compute an SVD decomposition of the forecast ensemble, the Eq. (66) can be written as

$$\boldsymbol{A}^a = \boldsymbol{A}\boldsymbol{X}_5, \tag{77}$$

$$= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{X}_5, \tag{78}$$

$$= \boldsymbol{U}\boldsymbol{X}_6. \tag{79}$$

Thus, it is possible to visualize the analysis as a combination of orthogonal singular vectors. This procedure may be useful, since it allows us to reject possible dependent ensemble members and possibly add new orthogonal members if these are needed. In particular, it can be used to examine how linearly independent the ensemble of model states is.

Some interesting conclusions which can be drawn are:

1. The covariances are only indirectly used to create the matrix $\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^T$, which includes covariances only between the observed variables at the locations of the observations. The actual covariances are never computed when the SVD algorithm in Section 4.3.2 is used, although they are used implicitly.
2. The analysis is not really computed as a combination of covariance functions. It is, in fact, computed as a combination of the forecast ensemble members. Each of these members can be considered as drawn from an infinite sample of dynamically consistent model states where the correct multivariate correlations are present in each ensemble member.
3. The covariances are important only for computing the best possible combination, i.e., the matrix $\boldsymbol{X}_5$. As long as the ensemble-based $\boldsymbol{X}_5$ is a good approximation, the accuracy of the final analysis will be determined by how well the error space spanned by the ensemble members represents the true error space of the model.

Clearly, from Eq. (72), the analysis becomes a combination of model states even if Eq. (68) is used for the actual computation, since these two equations are identical.

For a linear model a linear combination of model solutions is also a solution of the model. Thus, for a linear model, any choice of $\boldsymbol{X}_5$ will produce an ensemble of analyzed model states which is also a solution of the linear model.

If Eq. (68) is used for the computation of the analysis but with filtering applied to the covariance functions, one actually introduces spurious or nondynamical modes in the analysis. Based on these points, it is not wise to filter covariance functions, as has been proposed in a number of studies, e.g., by Mitchell (2001).

4.4 Local analysis

To avoid the problems associated with a large m, many operational assimilation schemes have made an assumption that only measurements located within a certain distance from a grid point will impact the analysis in this grid point. This allows for an algorithm where the analysis is computed grid point by grid point. Only a subset of observations which are located near the current grid point is used in the analysis for this particular grid point. This algorithm is approximative and it does not solve the original problem posed. Further, it is not clear how serious the approximation is.

Such an approach is not needed for handling a large m in the EnKF if the algorithm just described is used. However, there are other arguments for computing local analyses grid point by grid point. The analysis in the EnKF is computed in a space spanned by the ensemble members. This is a subspace which is rather small compared to the total dimension of the model state. Computing the analysis grid point by grid point implies that a small model state is solved for in a relatively large ensemble space. Further, the analysis will use a different combination of ensemble members for each grid point, and this also allows for a larger flexibility in the scheme to reach different model solutions.

For each horizontal grid point, we can now compute the corresponding $\boldsymbol{X}_5$ using only the selected measurements contributing to that particular grid point and update the ensemble for that particular grid point. The analysis at grid point (i,j), i.e., $\boldsymbol{A}^a_{(i,j)}$ then becomes:

$$\boldsymbol{A}^a_{(i,j)} = \boldsymbol{A}_{(i,j)}\boldsymbol{X}_{5,(i,j)} \tag{80}$$

$$= \boldsymbol{A}_{(i,j)}\boldsymbol{X}_5 + \boldsymbol{A}_{(i,j)}(\boldsymbol{X}_{5,(i,j)} - \boldsymbol{X}_5), \tag{81}$$

where $\boldsymbol{X}_5$ is the global solution while $\boldsymbol{X}_{5,(i,j)}$ becomes the solution for a local analysis corresponding to grid point (i,j) where only the nearest measurements are used in the analysis. Thus, it is possible to compute the global analysis first, and then add the corrections from the local analysis if these are significant.

The quality of the EnKF analysis is clearly connected to the ensemble size used. We expect that a larger ensemble is needed for the global analysis than the local analysis to achieve the same quality of the result. That is, in the global analysis a large ensemble is needed to properly explore the state space and to provide a consistent result for the global analysis which is as good as the local analysis. We expect this to be application-dependent. Note also that the use of a local analysis scheme is likely to introduce nondynamical modes, although the amplitudes of these will be small if a large enough influence radius is used when selecting measurements.

In dynamical models with large state spaces, the local analysis allows for the computation of a realistic analysis result while still using a relatively small ensemble of model states. This also relates to the discussions on localization and filtering of long range correlations by Mitchell et al. (2002).

4.5 Nonlinear measurement operators

The expression $\boldsymbol{D}' = \boldsymbol{D} - \boldsymbol{H}\boldsymbol{A}$ is just the difference between the ensemble of measurements and the ensemble of observed model states. If the observations are nonlinear functions of the model state, this matrix formulation using $\boldsymbol{H}$ becomes invalid. The traditional solution is to linearize and iterate. It is possible to augment the model state with a diagnostic variable which is the model prediction of the measurement. Thus, if $\boldsymbol{d} = \boldsymbol{h}(\psi,\ldots) + \epsilon$, then a new model state can be defined for each ensemble member as:

$$\widehat{\boldsymbol{\psi}}^T = [\psi^T, \boldsymbol{h}^T(\psi,\ldots)]. \tag{82}$$

By defining the new ensemble matrix as:

$$\widehat{\boldsymbol{A}} = (\widehat{\boldsymbol{\psi}}_1, \widehat{\boldsymbol{\psi}}_2, \ldots, \widehat{\boldsymbol{\psi}}_N) \in \Re^{\hat{n}\times N}, \tag{83}$$

with $\hat{n}$ being the n plus the number of measurement equivalents added to the original model state, the analysis can be written:

$$\boldsymbol{A}^a = \boldsymbol{A} + \boldsymbol{A}'\widehat{\boldsymbol{A}}'^T\widehat{\boldsymbol{H}}^T\left(\widehat{\boldsymbol{H}}\widehat{\boldsymbol{A}}'\widehat{\boldsymbol{A}}'^T\widehat{\boldsymbol{H}}^T + \Upsilon\Upsilon^T\right)^{-1}\boldsymbol{D}', \tag{84}$$

where the now linear innovations (with $\widehat{\boldsymbol{H}}$ being a direct and linear measurement functional) becomes

$$\boldsymbol{D}' = \boldsymbol{D} - \widehat{\boldsymbol{H}}\widehat{\boldsymbol{A}}. \tag{85}$$

From this expression, where the ensemble members have been augmented with the observation equivalent, we can compute the following using a linear (direct) measurement functional: the innovation $\boldsymbol{D}'$; the model-predicted error covariance of the observation's equivalents $\boldsymbol{H}\widehat{\boldsymbol{A}}'\widehat{\boldsymbol{A}}'^T\boldsymbol{H}^T$; and the covariance between the observations and all prognostic model variables from $\boldsymbol{A}'\widehat{\boldsymbol{A}}'^T\widehat{\boldsymbol{H}}^T$.

The analysis is a combination of model-predicted error covariances between the observation equivalents $\boldsymbol{h}(\psi,\ldots)$ and all other model variables. Thus, we have a fully multivariate analysis scheme.

4.6 Assimilation of "nonsynoptic" measurements

In some cases, measurements occur with high frequency in time. An example is along-track satellite data. It is not

practical to perform an analysis every time there is a measurement. Further, the normal approach of assimilating, at one time instant, all data collected within a time interval, is not optimal. Based on the theory from Evensen and van Leeuwen (2000), it is possible to assimilate the nonsynoptic measurements at one time instant by exploiting the time correlations in the ensemble. Thus, a measurement collected at a previous time allows for the computation of the $\boldsymbol{HA}$ at that time and thereby also the innovations. By treating these as augmented model variables, Eq. (84) can again be used but with the $\boldsymbol{h}(\psi, \ldots)$ now denoting the measurements collected at earlier times.

5 Numerical implementation of the EnKF

The algorithm as explained in the previous sections provides an optimal approach for computing the EnKF analysis. The following provides a basis explaining the implementation of the EnKF analysis scheme. It assumes access to the BLAS and EISPACK libraries, where highly optimized numerical subroutines are available for most computer systems and which can be obtained for free from the archive at `www.netlib.no`.

5.1 Storing the ensemble on disk

For most practical applications one will not want to keep the whole ensemble in memory during the ensemble integrations. Rather, an approach where ensemble members are kept in a file residing on disk is convenient. This allows for the system to read a particular member from file, integrate it forward in time, and then store it on disk again following the integration. An approach where each member is stored in a record in a direct Fortran file is most convenient. This allows us to read and write specific records containing individual ensemble members.

5.2 Analysis implementation

The algorithm for the analysis exploits that we can compute once and store all innovations, measurement perturbations and the measurements of the ensemble. Thus, we start with the following:

1. Read the whole ensemble forecast into $\boldsymbol{A}$.
2. Compute the matrix $\boldsymbol{HA}$.
3. Compute the measurement perturbations Υ.
4. Compute the innovations $\boldsymbol{D}'$.
5. Compute $\overline{\boldsymbol{HA}}$ and subtract it from $\boldsymbol{HA}$ to get $\boldsymbol{HA}'$ (requires $\boldsymbol{H}$ to be linear).

The following subroutine can then be used without any modification to compute the analysis for all kinds of model states.

```
subroutine analysis (A, D, E, S, ndim, nrens, nrobs)
! Computes the analysed ensemble in the EnKF
! Written by G. Evensen (Geir.Evensen@nersc.no)
! This routine uses subroutines from BLAS and EISPACK
! and calls the additional multiplication routine multa.
 use m_multa
 implicit none

! Dimension of model state
   integer, intent(in) :: ndim
! Number of ensemble members
   integer, intent(in) :: nrens
! Number of observations
   integer, intent(in) :: nrobs
! Ensemble matrix
   real, intent(inout) :: A(ndim, nrens)
! Matrix holding innovations
   real, intent(in)    :: D(nrobs, nrens)
! Matrix holding HA'
   real, intent(in)    :: S(nrobs, nrens)
! Matrix holding observation perturbations
real, intent(in)       :: E(nrobs, nrens)

! Local variables
  real, allocatable, dimension (:,:) :: &
   X1, X2, U, X4, Reps
  real, allocatable, dimension (:)   :: &
   sig, work
  real ES(nrobs, nrens), X3(nrobs, nrens), V(nrens, nrens)
  real sigsum, sigsum1
  integer ierr, nrsigma, i, j, lwork, nrmin, iblkmax

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Do nothing if only one measurement
   if (nrobs == 1) then
      print *,' analysis: no support for nrobs = 1'
      return
   endif
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Minimum of nrobs and nrens
   nrmin = min (nrobs, nrens)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Compute HA' + E
   ES = S + E
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
! Compute SVD of HA' + E-> U and sig, using Eispack
  allocate (U(nrobs, nrmin))
  allocate (sig(nrmin))
  lwork = 2*max(3*nrens + nrobs, 5*nrens)
  allocate (work(lwork))
  sig = 0.0
  call dgesvd ('S', 'N', nrobs, nrens, ES, nrobs, sig, &
                  U, nrobs, V, nrens, work, lwork, ierr)
  deallocate(work)
  if (ierr/ = 0)then
    print *, 'ierr from call dgesvd =', ierr
    stop
  endif

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Convert to eigenvalues
  do i = 1, nrmin
    sig (i) = sig (i) **2
  enddo

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Compute number of significant eigenvalues
  sigsum = sum(sig (1 : nrmin))
  sigsum1 = 0.0
  nrsigma = 0
  do i = 1, nrmin
    if (sigsum1/sigsum < 0.999) then
      nrsigma = nrsigma + 1
      sigsum1 = sigsum1 + sig(i)
      sig(i) = 1.0/sig(i)
    else
      sig(i : nrmin) = 0.0
      exit
    endif
  enddo

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Compute X1
  allocate (X1(nrmin, nrobs))
  do j = 1, nrobs
  do i = 1, nrmin
    X1(i, j) = sig(i) * U(j, i)
  enddo
  enddo
  deallocate(sig)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Compute X2 = X1 * D
  allocate (X2(nrmin, nrens))
  call dgemm('n' , 'n' , nrmin, nrens, nrobs, 1.0, X1, &
              nrmin, D, nrobs, 0.0, X2, nrmin)
  deallocate(X1)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Compute X3 = U * X2
  call dgemm('n', 'n', nrobs, nrens, nrmin, 1.0, U, &
              nrobs, X2, nrmin, 0.0, X3, nrobs)
  deallocate(U)
  deallocate(X2)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Compute final analysis
  if(2 * ndim * nrobs > nrens * (nrobs + ndim)) then
!  Case with nrobs 'large'
!    Compute X4 = (HA')^T * X3
    allocate (X4(nrens, nrens))
    call dgemm('t', 'n', nrens, nrens, nrobs, 1.0, &
                S, nrobs, X3, nrobs, 0.0, X4, nrens)
!    Compute X5 = X4 + I (stored in X4)
    do i = 1, nrens
      X4(i,i) = X4(i,i) + 1.0
    enddo
!    Compute A = A * X5
    iblkmax = min (ndim, 200)
    call multa(A, X4, ndim, nrens, iblkmax)
    deallocate (X4)
  else
!  Case with nrobs 'small'
!    Compute representers Reps = A' * S^T
    allocate (Reps(ndim, nrobs))
    call dgemm('n', 't', ndim, nrobs, nrens, 1.0, A, &
                ndim, S, nrobs, 0.0, Reps, ndim)
!    Compute A = A + Reps * X3
    call dgemm('n', 'n', ndim, nrens, nrobs, 1.0, &
                Reps, ndim, X3, nrobs, 1.0, A, ndim)
    deallocate(Reps)
  endif
end subroutine analysis
```

5.3 Final update

The most demanding step in the EnKF analysis is the final step when evaluating the analysis ensemble from Eq. (72), i.e.,

$$\boldsymbol{A} = \boldsymbol{A}\boldsymbol{X}_5, \tag{86}$$

with

$$\boldsymbol{X}_5 = \boldsymbol{I} + \boldsymbol{X}_4 \in \Re^{N\times N}. \tag{87}$$

Here the largest matrix to be held in memory is the ensemble matrix $\boldsymbol{A} \in \Re^{n\times N}$. Further, the number of floating point operations (a multiply and add) is nN^2 which is likely to be several orders of magnitude more than for the previous steps in the algorithm.

This matrix multiplication can easily be computed while overwriting row by row of $\boldsymbol{A}$ using the subroutine `multa` listed below, which requires only one copy of the ensemble to be kept in memory. This subroutine has been found to perform the multiplication very efficiently. It uses optimized BLAS routines and includes a block representation where only a small part of the model state needs to be held as an additional copy in memory during the multiplication.

```
module m_multa
contains
subroutine multa (A, X, ndim, nrens, iblkmax)
implicit none
integer, intent(in) :: ndim
integer, intent(in) :: nrens
integer, intent(in) :: iblkmax
real, intent(in) :: X(nrens, nrens)
real, intent(inout) :: A(ndim, nrens)
real v(iblkmax, nrens) ! Automatic work array

integer ia, ib
do ia = 1, ndim, iblkmax
  ib = min(ia + iblkmax - 1, ndim)
  v(1 : ib - ia + 1, 1 : nrens) = A(ia : ib, 1 : nrens)
  call dgemm('n', 'n', ib - ia + 1, nrens, nrens, &
             1.0, v(1, 1), iblkmax, &
             X(1, 1), nrens, &
             0.0, A(ia, 1), ndim)
enddo
end subroutine multa
end module m_multa
```

5.3.1 Remark 1

Note that this routine does not care about the order in which the elements in `A` are stored for each ensemble member. Thus, in the call to `multa`, `A` can be a multi dimensional matrix e.g., `A(nx,ny,nz,nrens)` holding an ensemble of a univariate three-dimensional model state. A multivariate model state can be stored in a structure or type declaration, and still be input to `multa`.

5.3.2 Remark 2

In principle, the multiplication has a serious drawback caused by the stride `ndim` copies. Here the routine relies on BLAS for the inner matrix multiplication, since the BLAS routines have already been designed to optimize cache performance. The variable `iblkmax` is used only for storage considerations and a typical value of 200 seems to work well. This routine also opens for a possible block representation of the model state.

5.4 A block algorithm for large ensemble matrices

It is still possible to use the EnKF even if the whole ensemble does not fit in memory. In this case, a block algorithm can be used for the final update, using additional reads and writes to file. The algorithm goes as follows:

1. Read each individual ensemble member into a vector one at the time while computing and storing the columns of $\boldsymbol{HA}$.
2. Compute the measurement perturbations Υ.
3. Compute the innovations $\boldsymbol{D}'$.
4. Compute $\overline{\boldsymbol{HA}}$ and subtract it from $\boldsymbol{HA}$ to obtain $\boldsymbol{HA}'$.

Use the algorithm described in the `analysis` subroutine to compute $\boldsymbol{X}_5$. So far, we have only kept one full model state in memory at the time. It remains to solve the Eq. (86). Using the block algorithm just discussed, it is possible to perform this computation without keeping all of the ensemble in memory at once. A proposed strategy is to store the ensemble in several files, say one file for the temperature, one for the salinity, etc. Then the analysis can be performed sequentially on the individual blocks, at the cost of one additional read and write of the whole ensemble.

Appendix A: Consistency checks on error statistics

The EnKF provides error statistics for the results. To validate the predicted error statistics it is possible to compare statistics computed from the innovation sequence with the predicted error statistics.

If the model forecast is written as:

$$\psi^f = \psi^t + \boldsymbol{q}, \tag{88}$$

i.e., it is given as the truth plus an error, and the measurements are written as:

$$\boldsymbol{d} = \boldsymbol{H}\psi^t + \epsilon, \tag{89}$$

the innovation becomes:

$$\boldsymbol{d} - \boldsymbol{H}\psi^f = \epsilon - \boldsymbol{H}\boldsymbol{q}. \tag{90}$$

By squaring this equation and taking the expectation we obtain the expression

$$\overline{(\boldsymbol{d} - \boldsymbol{H}\psi^f)(\boldsymbol{d} - \boldsymbol{H}\psi^f)^T} = \boldsymbol{R} + \boldsymbol{H}\boldsymbol{P}^f\boldsymbol{H}^T, \tag{91}$$

where correlations between the forecast error and the measurement error have been neglected.

Thus, it is possible to compute the variance of the innovation sequence in time, subtract the measurement variance and compare this with the predicted error variance from the ensemble. This provides a solid consistency test on the prescribed error statistics used in the EnKF.

Appendix B: Ensemble Optimal Interpolation (EnOI)

Traditional optimal interpolation (OI) schemes have estimated or prescribed covariances using an ensemble of model states which has been sampled during a long time integration. Normally, the estimated covariances are fitted to simple functional forms which are used uniformly throughout the model grid.

Based on the discussion in this paper it is natural to derive an OI scheme where the analysis is computed in the space spanned by a stationary ensemble of model states sampled, e.g., during a long time integration. This approach is denoted Ensemble OI (EnOI).

The EnOI analysis is computed by solving an equation similar to (54) but written as:

$$\psi^a = \psi + \alpha \boldsymbol{A}'\boldsymbol{A}'^T\boldsymbol{H}^T\left(\alpha \boldsymbol{H}\boldsymbol{A}'\boldsymbol{A}'^T\boldsymbol{H}^T + \Upsilon\Upsilon^T\right)^{-1}(\boldsymbol{d} - \boldsymbol{H}\psi). \tag{92}$$

The analysis is now computed for only one single model state, and a parameter $\alpha \in (0,1]$ is introduced to allow for different weights on the ensemble versus measurements. Naturally, an ensemble consisting of model states sampled over a long time period will have a climatological variance which is too large to represent the actual error in the model forecast, and α is used to reduce the variance to a realistic level.

The practical implementation introduces α in Eq.(59), which is now written as:

$$\sqrt{\alpha}\boldsymbol{H}\boldsymbol{A}' + \Upsilon = \boldsymbol{U}\Sigma\boldsymbol{V}^T, \tag{93}$$

and the coefficient matrix $\boldsymbol{X}_4$ in Eq.(65) is further scaled with α before $\boldsymbol{X}_5$ is computed.

The EnOI method allows for the computation of a multivariate analysis in dynamical balance, just like the EnKF. However, a larger ensemble may be useful to ensure that it spans a large enough space to properly represent the correct analysis.

The EnOI can be an attractive approach to save computer time. Once the stationary ensemble is created, only one single model integration is required in addition to the analysis step where the final update cost is reduced to $\mathcal{O}(nN)$ floating point operations because only one model state is updated. The method is numerically extremely efficient but it will always provide a suboptimal solution compared to the EnKF. In addition it does not provide consistent error estimates for the solution.

Appendix C: Assimilation of in situ measurements

The direct assimilation of in situ observations such as temperature and salinity profiles is problematic in ocean models unless both temperature and salinity are known simultaneously or the correct temperature-salinity correlation is known. Thus, using simple assimilation schemes, it is not known, a priori, how to update the water-mass properties in a consistent manner, see, e.g., Troccoli et al. (2002) and Thacker and Esenkov (2002).

From the interpretation of the EnKF analysis as a combination of valid model states, and the discussion on nonlinear measurement functionals in the EnKF, it is possible to compute a consistent analysis even if only temperature (or salinity) is observed. The algorithm applies a definition of a measurement functional which interpolates the model temperature (or salinity) to the measurement location in depth. The model state is then augmented with the observation equivalents for each independent in situ measurement. Innovations and covariances between the measurement equivalents can then be evaluated and the standard analysis equations can be used to compute the analysis. This approach ensures that the model update in the vertical and horizontal is performed consistently with the error statistics predicted by the ensemble.

In order to obtain a variable's value at a specific depth, an interpolation algorithm is needed. We use a second-order spline to interpolate in the vertical. It is important to note that when interpolating values between different layers the interpolating spline should not pass exactly through the mean of the variable at the center of each layer. Instead, a criterion is used where the mean value computed by integrating the spline function across the layer is equal to the mean of the variable in that layer. The details of the algorithm follows.

Appendix C.1: Upper layer

Layer one is divided into upper and lower parts where the spline polynomial, used to represent the variable to be interpolated is defined as:

$$f_1(x) = \begin{cases} c_0 & \text{for } x \in [0, \frac{1}{2}h_1) \\ a_1x^2 + b_1x + c_1 & \text{for } x \in [\frac{1}{2}h_1, h_1]. \end{cases} \tag{94}$$

Here h_i is the location of the lower interface of layer i. Conditions are specified at $x = \frac{1}{2}h_1$ for continuity of the function and the derivative, i.e.,

$$f_1\left(\frac{1}{2}h_1\right) = c_0, \tag{95}$$

and

$$\left.\frac{\partial f_1(x)}{\partial x}\right|_{\frac{1}{2}h_1} = 0, \tag{96}$$

and in addition the integral over layer 1 should satisfy

$$\frac{1}{h_1}\int_0^{h_1} f_1(x) = c_0\frac{1}{2} + a_1\frac{7}{24}h_1^2 + b_1\frac{3}{8}h_1 + c_1\frac{1}{2} = \bar{u}_1, \tag{97}$$

with $\bar{u}_1$ being the model-predicted layer variable in layer one.

Appendix C.2: Interior layers

Within each interior layer, i, a function of the form

$$f_i(x) = a_i x^2 + b_i x + c_i, \tag{98}$$

is used to represent the model variables. For each interior layer there are three conditions which determine the three unknowns in each layer, i.e., continuity at layer interfaces

$$f_i(h_{i-1}) = f_{i-1}(h_{i-1}), \tag{99}$$

continuity of derivatives at layer interfaces

$$\left.\frac{\partial f_i(x)}{\partial x}\right|_{h_{i-1}} = \left.\frac{\partial f_{i-1}(x)}{\partial x}\right|_{h_{i-1}}, \tag{100}$$

and a condition for the mean of the variable becomes after some manipulations

$$\frac{1}{h_i - h_{i-1}} \int_{h_{i-1}}^{h_i} f_i(x)dx = a_i \frac{1}{3}\left(h_{i-1}^2 + h_{i-1}h_i + h_i^2\right) + b_i \frac{1}{2}(h_i + h_{i-1}) + c_i = \bar{u}_i. \tag{101}$$

Appendix C.3: Closing the system

A final condition is obtained by setting the variable at the sea floor equal to the mean of the variable in the bottom layer,

$$f_k(h_k) = \bar{u}_k. \tag{102}$$

Thus, the system is closed.

Appendix d: Ensemble Kalman Smoother (EnKS)

In light of the discussion in this paper it is also possible to derive an efficient implementation of the EnKS. The EnKS, as described in Evensen and van Leeuwen (2000), updates the ensemble at prior times every time new measurements are available. The update exploits the space–time correlations between the model forecast at measurement locations and the model state at a prior time. It allows for a sequential processing of the measurements in time. Thus, every time a new set of measurements becomes available, the ensemble at the current and all prior times can be updated.

Similar to the analysis Eq. (54) the analysis for a prior time t' which results from the introduction of a new measurement vector at time $t > t'$ can be written as

$$\boldsymbol{A}^a(t') = \boldsymbol{A}(t') + \boldsymbol{A}'(t')\boldsymbol{A}'^T(t)\boldsymbol{H}^T \left(\boldsymbol{H}\boldsymbol{A}'(t)\boldsymbol{A}'^T(t)\boldsymbol{H}^T + \Upsilon\Upsilon^T\right)^{-1}\boldsymbol{D}'(t). \tag{103}$$

This equation is updated repetitively every time a new set of measurements are introduced at future times t.

The EnKS analysis can best be computed using the formulation discussed in the previous sections, and in particular using the definition of $\boldsymbol{X}_5$ in Eq. (87). It is easily seen that the matrix of coefficients $\boldsymbol{X}_5(t)$ corresponding to the measurements at time t, is also used on the analysis ensemble at the prior times t' to update the smoother estimate at time t'.

Thus, the smoother estimate at a time t' where $t_{i-1} \le t' < t_i \le t_k$, using data from the future data times, i.e., $(t_i, t_{i+1}, \ldots, t_k)$, is just

$$\boldsymbol{A}^a_{\mathrm{EnKS}}(t') = \boldsymbol{A}_{\mathrm{EnKF}}(t') \prod_{j=i}^{k} \boldsymbol{X}_5(t_j). \tag{104}$$

As long as the previous ensemble files have been stored, it is straightforward to update them with new information every time a new set of measurements is available and the matrix $\boldsymbol{X}_5$ corresponding to these measurements has been computed. This discussion has assumed that a global analysis is used. The local analysis becomes a little less practical since there is an $\boldsymbol{X}_5$ matrix for each grid point.

The product in Eq. (104) has an important property. The multiplication of the ensemble with $\boldsymbol{X}_5$ will always result in a new ensemble with a different mean and a smaller variance. Thus, each consecutive update through the repetitive multiplication in (104) will lead to slight reduction of variance and slight change of mean. Eventually, there will be a convergence with only negligible updates of the ensemble when measurements are taken further into the future than the actual decorrelation time.

Appendix E: Generating pseudorandom fields

Here a procedure is given which can be used to compute smooth pseudo random fields with mean equal to zero, variance equal to one, and a specified covariance which determines the smoothness of the fields. The algorithm follows the presentation in the appendix of Evensen (1994a), and additional details and explanations are given by Natvik (2001).

Let $q = q(x, y)$ be a continuous field, which may be described by its Fourier transform

$$q(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \widehat{q}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k}. \tag{105}$$

Now, we are using an $N \times M$ grid. Further, we define $\mathbf{k} = (\kappa_l, \gamma_p)$, where l and p are counters and κ_l and γ_p are wave numbers in the N and M directions, respectively. We now obtain a discrete version of Eq. (105),

$$q(x_n, y_m) = \sum_{l,p} \widehat{q}(\kappa_l, \gamma_p) e^{i(\kappa_l x_n + \gamma_p y_m)} \Delta\mathbf{k}, \tag{106}$$

where $x_n = n\Delta x$ and $y_m = m\Delta y$. For the wavenumbers, we have

$$\kappa_l = \frac{2\pi l}{x_N} = \frac{2\pi l}{N\Delta x}, \tag{107}$$

$$\gamma_P = \frac{2\pi p}{y_M} = \frac{2\pi p}{M\Delta y}, \tag{108}$$

$$\Delta\mathbf{k} = \Delta\kappa\Delta\gamma = \frac{(2\pi)^2}{NM\Delta x\Delta y}. \tag{109}$$

We define (assume) the following form of $\widehat{q}(\mathbf{k})$:

$$\widehat{q}(\kappa_l,\gamma_p) = \frac{c}{\sqrt{\Delta\mathbf{k}}} e^{-(\kappa_l^2+\gamma_p^2)/\sigma^2} e^{2\pi i\phi_{l,p}}, \tag{110}$$

where $\phi_{l,p} \in [0,1]$ is a random number which introduces a random phase shift. (The exponential function may be written as a sum of sine and cosine terms). Note that increasing wavenumbers κ_l and γ_p will give an exponentially decreasing contribution to the expression above. Now, Eq. (110) may be inserted into Eq. (106), and we obtain:

$$q(x_n,y_m) = \sum_{l,p} \frac{c}{\sqrt{\Delta\mathbf{k}}} e^{-(\kappa_l^2+\gamma_p^2)/\sigma^2} e^{2\pi i\phi_{l,p}} e^{i(\kappa_l x_n+\gamma_p y_m)}\Delta\mathbf{k}. \tag{111}$$

We want Eq. (111) to produce real fields only. Thus, when the summation over l,p is performed, all the imaginary contributions must add up to zero. This is satisfied whenever

$$\widehat{q}(\kappa_l,\gamma_p) = \widehat{q}^*(\kappa_{-l},\gamma_{-p}), \tag{112}$$

where the star denotes complex conjugate, and

$$\mathrm{Im}\,\widehat{q}(\kappa_0,\gamma_0) = 0. \tag{113}$$

The formula (111) can be used to generate an ensemble of pseudorandom fields with a specific covariance determined by the parameters c and σ. An expression for the covariance is given by:

$$\overline{q(x_1,y_1)q(x_2,y_2)} = \sum_{l,p,r,s} \widehat{q}(\kappa_l,\gamma_p)\widehat{q}(\kappa_r,\gamma_s) e^{i(\kappa_l x_1+\gamma_p y_1+\kappa_r x_2+\gamma_s y_2)}(\Delta\mathbf{k})^2. \tag{114}$$

By using Eq. (112), and by noting that the summation goes over both positive and negative r and s, we may insert the complex conjugate instead, i.e.,

$$\begin{aligned}\overline{q(x_1,y_1)q(x_2,y_2)} &= \sum_{l,p,r,s} \widehat{q}(\kappa_l,\gamma_p)\widehat{q}^*(\kappa_r,\gamma_s) e^{i(\kappa_l x_1-\kappa_r x_2+\gamma_p y_1-\gamma_s y_2)}(\Delta\mathbf{k})^2 \\ &= \sum_{l,p,r,s} \Delta\mathbf{k}c^2 e^{-(\kappa_l^2+\gamma_p^2+\kappa_r^2+\gamma_s^2)/\sigma^2} e^{2\pi i(\phi_{l,p}-\phi_{r,s})} \\ &\quad\times e^{i(\kappa_l x_1-\kappa_r x_2+\gamma_p y_1-\gamma_s y_2)}.\end{aligned} \tag{115}$$

We assume that the fields are δ- correlated in wave space. That is, we assume that there is a distance dependence only (isotropy), and we may set $l = r$ and $p = s$, and the above expression becomes

$$\overline{q(x_1,y_1)q(x_2,y_2)} = \Delta\mathbf{k}c^2 \sum_{l,p} e^{-2(\kappa_l^2+\gamma_p^2)/\sigma^2} e^{i\left[\kappa_l(x_1-x_2)+\gamma_p(y_1-y_2)\right]}. \tag{116}$$

From this equation, the variance at (x,y) is

$$\overline{q(x,y)q(x,y)} = \Delta\mathbf{k}c^2 \sum_{l,p} e^{-2(\kappa_l^2+\gamma_p^2)/\sigma^2}. \tag{117}$$

Now, we require the variance to be equal to 1. Further, we define a decorrelation length r_h, and we require the covariance corresponding to r_h to be equal to e^{-1}. For the variance, we get the equation:

$$1 = \Delta\mathbf{k}c^2 \sum_{l,p} e^{-2(\kappa_l^2+\gamma_p^2)/\sigma^2}, \tag{118}$$

which means that

$$c^2 = \frac{1}{\Delta\mathbf{k}\sum_{l,p} e^{-2(\kappa_l^2+\gamma_p^2)/\sigma^2}}. \tag{119}$$

If we let $x_1 - x_2 = r_h$ and $y_1 - y_2 = 0$, we must have a covariance equal to e^{-1} between these points, i.e.,

$$\begin{aligned}e^{-1} &= \Delta\mathbf{k}c^2 \sum_{l,p} e^{-2(\kappa_l^2+\gamma_p^2)/\sigma^2} e^{i\kappa_l r_h} \\ &= \Delta\mathbf{k}c^2 \sum_{l,p} e^{-2(\kappa_l^2+\gamma_p^2)/\sigma^2} \cos(\kappa_l r_h).\end{aligned} \tag{120}$$

By inserting for c^2 from Eq. (119), we obtain:

$$e^{-1} = \frac{\sum_{l,p} e^{-2(\kappa_l^2+\gamma_p^2)/\sigma^2}\cos(\kappa_l r_h)}{\sum_{l,p} e^{-2(\kappa_l^2+\gamma_p^2)/\sigma^2}}. \tag{121}$$

This is a nonlinear scalar equation for σ, which may be solved using some numerical routine. One can thereafter find a value for c from Eq. (119).

Once the values for c and σ have been determined, Eq. (111) may be used to create an ensemble of pseudorandom fields with variance 1 and covariance determined by the decorrelation length r_h. An efficient approach for finding the inverse transform in Eq. (111) is to apply a two-dimensional fast Fourier transform (FFT). The inverse FFT is calculated on a grid which is a few characteristic lengths larger than the computational domain to ensure nonperiodic fields in the subdomain corresponding to the computational domain (Evensen, 1994a).

To summarize, we are now able to generate (sample) two-dimensional pseudorandom fields with variance equal to 1 and a prescribed covariance (isotropic as a function of grid indices). The simple formulas used in Section 2 (i.e., using Eq. 28 with a choice of α) can be used to introduce correlations between the fields.

Appendix F: An example with a scalar model

A simple example is now presented to illustrate some of the properties of the EnKF and EnKS. There are already a large number of applications of the EnKF using different physical models, as cited in Section 2. Thus, in the following example the focus will be on the stochastic behaviour of the EnKF and EnKS rather than the impact of a dynamical model on the evolution of error statistics.

The model considered is the linear scalar version of the model described by Eq. (43), with $f(\psi) = \psi$, i.e.,

$$\begin{pmatrix} q_k \\ \psi_k \end{pmatrix} = \begin{pmatrix} \alpha q_{k-1} \\ \psi_{k-1} + \sqrt{\Delta t}\sigma\rho q_k \end{pmatrix} + \begin{pmatrix} \sqrt{1-\alpha^2} w_{k-1} \\ 0 \end{pmatrix}. \tag{122}$$

Two experiments will be discussed, which both use the following parameter settings:

1. The time interval is from 0 to 10 time units;
2. the ensemble size is 1000, which is large enough to eliminate any visible effects of using a finite ensemble;
3. the time step is $dt = 0.1$, which splits the time interval into 100 subintervals;
4. the initial ensemble is sampled from $\mathcal{N}(0.0, 1.0)$, it has mean zero and variance equal to one;
5. the model errors are sampled from $\mathcal{N}(0.0, 1.0)$, indicating that the model error variance will increase with 1.0 when the ensemble is integrated over one time unit;
6. the measurements are sampled from $\mathcal{N}(0.0, 0.5)$;
7. there are five measurement times distributed equally over the time interval.

The two experiments use different values for the coefficient α which denote the time correlation of the model errors. In case A, $\alpha = 0.00$, which implies white noise, while in case B a strong time correlation is imposed using $\alpha = 0.95$, which results in approximately 2 time units as the decorrelation time from Eq. (32). The factor ρ is computed using the formula (42) to ensure that the two cases have similar error variance growth in the stochastic model.

In all the cases both the EnKF and EnKS solutions are computed, and the procedure for estimation of model noise discussed in Section 4.2.2 is used.

The EnKF and EnKS results from Case A are shown in Fig. 1. The thick line shows the ensemble mean which represents the EnKF estimate. At the measurement locations, represented by the circles, there are discontinuities in the estimate due to the analysis updates. During the integration between the measurement locations the ensemble mean satisfies the dynamical part of the model equation, i.e., the time derivative of the solution is zero.

The two dotted lines represent the ensemble mean with the ensemble standard deviation respectively added and subtracted and thus indicate the ensemble spread around the mean. The ensemble standard deviation is reduced at every measurement time, and increases according to the stochastic forcing term during the integration between the measurements.

The thick dashed line is the EnKS solution, which is computed using the procedure outlined in Appendix D. Clearly, this provides a continuous curve which is a more realistic estimate than the EnKF solution. Note that, due to the use of white noise this curve will have discontinuous time derivatives at the measurement locations, a property also found in the representer solutions by Bennett (1992, 2002) when white model errors are used.

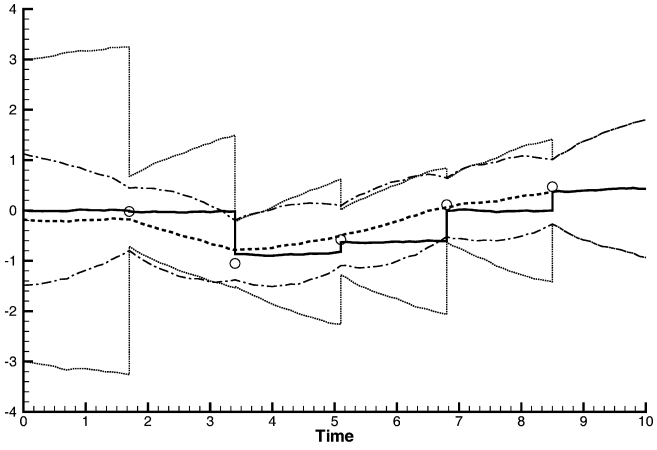

Fig. 1 Case A. Results using $\alpha = 0.0$. The *thick solid and dashed curves* are, respectively, the EnKF and EnKS estimates, while the *thin dotted and dash-dot curves* represent, respectively, the ensemble standard deviations around the EnKF and the EnKS estimates. The *circles* denote measurements

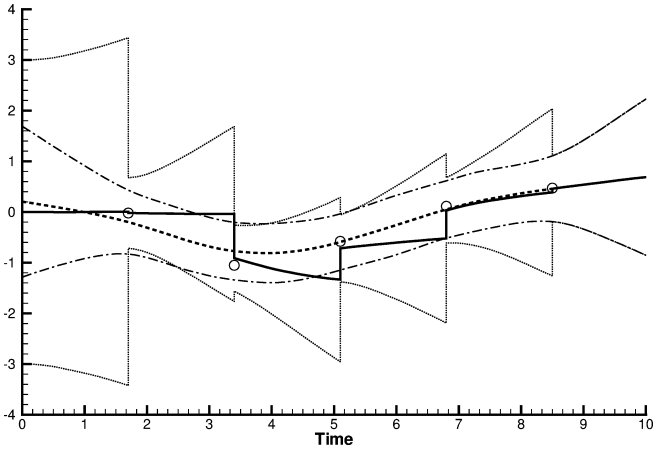

Fig. 2 Case B. Results using $\alpha = 0.95$. The *thick solid and dashed curves* are respectively, the EnKF and EnKS estimates, while the *thin dotted and dash-dot curves* represent, respectively, the ensemble standard deviations around the EnKF and the EnKS estimates. The *circles* denote measurements

The thin dash-dot lines indicate the ensemble standard deviation for the EnKS. Clearly, there is an impact backward in time from the measurements and the overall error estimate is smaller than for the EnKF. The minimum errors are found at the measurement locations as expected, and after the last measurement the EnKF and EnKS solutions are identical, in agreement with the theory from Evensen and van Leeuwen (2000).

In Fig. 2 the results from case B are presented. We have used the same format and line styles as was used in Fig. 1. There are several features to note from this experiment.

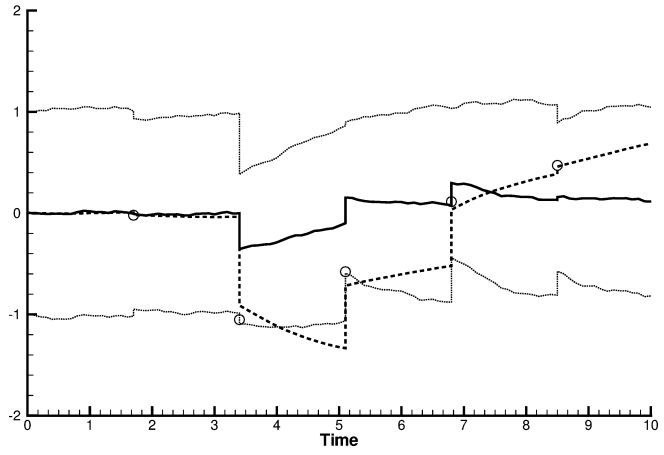

Fig. 3 Case B. The *thick solid line* is the EnKF estimate of the model noise. The *dotted lines* show the standard deviation of the model noise ensemble around the EnKF estimate of the model noise. The *dashed line* is the EnKF estimate as shown in Fig. 2

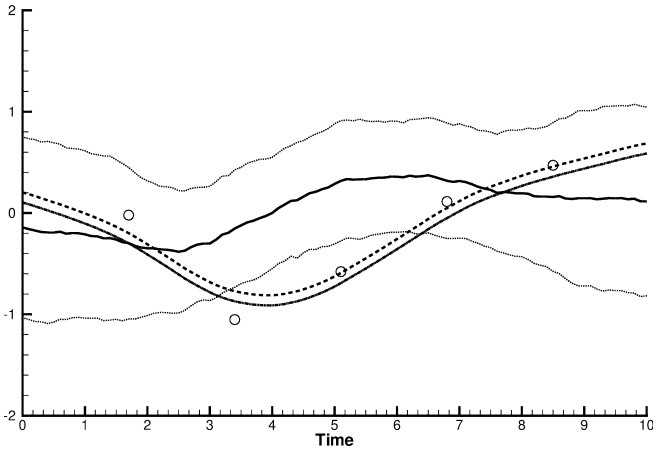

Fig. 4 Case B. The *thick solid line* is the EnKS estimate of the model noise. The *dotted lines* show the standard deviation of the model noise ensemble around the EnKS estimate of the model noise. The *dashed line* is the EnKS estimate as shown in Fig. 2. The *dash-dot-dot line* is the EnKS estimate computed from the forward model (123) but with a number 0.1 subtracted from the result to make it possible to distinguish the two curves

The EnKF solution now sometimes shows a positive or negative trend during the integration between the measurements. This is caused by the assimilation updates of the model noise which introduces a "bias" in the stochastic forcing. An explanation for this can be found by examining Fig. 3 which plots the EnKF solution as the dashed line, the EnKF estimate for the model noise as the thick solid line, and the standard deviation of the model noise is plotted as the dotted lines. It is clearly seen that the model noise is being updated at the assimilation steps, e.g., the second measurement indicates that the solution is around -1. This leads to an update of the model ensemble toward the measurement, as well as an introduction of a negative bias in the system noise. This is the negative bias which previously should have been used in the model to achieve a better prediction of this particular measurement. In the continued integration this bias starts working until it is corrected to become a positive bias at the next assimilation step. Note that during the integration between the measurements the bias slowly relaxes back toward zero as was discussed in Section 4.2.2.

The EnKS solution in case B is smoother than in case A and there are no longer discontinuous time derivatives at the measurement locations. Further, the standard deviation for the EnKS is smoother and indicates that the impact of the measurements is carried further backward in time.

The estimated EnKS system noise is presented as the thick solid line in Fig. 4 and also here the time derivatives are continuous at the measurement locations. In fact, this estimated model noise is the forcing needed to reproduce the EnKS solution when a single model is integrated forward in time starting from the initial condition estimated by the EnKS. That is, the solution of

$$\begin{aligned} \psi_k &= \psi_{k-1} + \sqrt{\Delta t}\sigma\rho\hat{q}_k \\ \psi_0 &= \hat{\psi}_0 \end{aligned} \tag{123}$$

with $\hat{q}_k$ and $\hat{\psi}_0$ being the EnKS estimated model noise and initial condition, respectively, will exactly reproduce the EnKS estimate.

This is illustrated by the two similar curves in Fig. 4, i.e., the dashed curve which is the EnKS estimate, and the dash-dot-dot curve which is the solution of the model forced by the estimated EnKS model noise (note that the curves are nearly identical and a number 0.1 was subtracted to make it easier to distinguish the curves). Obviously, the estimated model noise is the same as is computed and used in the forward Euler Lagrange equation in the representer method (Bennett, 1992, 2002). This points to the similarity between the EnKS and the representer method, which for linear models will give identical results when the ensemble size becomes infinite.

Appendix G: Bias and parameter estimation

The estimation of poorly known model parameters or biases in dynamical models has been discussed by, e.g., Evensen at al. (1998) and the references cited therein. The following examples will illustrate how the EnKF and EnKS may be used for parameter and bias estimation.

Note first, that the distinction between model bias and time correlated model errors is not clear. As an example, one can envisage an ocean model which overestimates a variable during summer and underestimates it during winter. This could happen if a constant is used to represent a process which changes slowly on the seasonal time scale. In a multiyear simulation, the poor representation of this process should be interpreted as a time-correlated model error. However, if the model is used only for short simulations, one could also consider

this error to be a model bias since it appears to be constant on short time scales.

The following model system based on Eq. (122) will now be examined in two examples to illustrate the impact of a model bias,

$$\begin{pmatrix} q_k \\ \beta_k \\ \psi_k \end{pmatrix} = \begin{pmatrix} \alpha q_{k-1} \\ \beta_{k-1} \\ \psi_{k-1} + (\eta + \beta_k)\Delta t + \sqrt{\Delta t}\sigma\rho q_k \end{pmatrix} + \begin{pmatrix} \sqrt{1-\alpha^2} w_{k-1} \\ 0 \\ 0 \end{pmatrix}. \quad (124)$$

In the dynamical model we have now included a "trend term", η, which states that the model solution as predicted by the deterministic model should have the slope η. In the examples below $\eta = 1.0$, while the measurements are simulated under the assumption that $\eta = 0.0$. We have also included an unknown bias parameter β_k, which will be estimated to correct for the model bias introduced by the trend term. An additional equation is introduced for the bias β_k, stating that it should be stationary in time.

Two examples will be discussed next. In both of them we set the following parameters in addition to the ones used in cases A and B:

1. The trend term $\eta = 1.0$, i.e., the model always predicts a slope equal to one;
2. the initial condition is $\psi_0 \in \mathcal{N}(2.0, 1.0)$ (2.0 is chosen instead of 0.0 to make it easier to interpret the plots);
3. the measurements are sampled from $\mathcal{N}(2.0, 0.5)$;
4. the model error has a time correlation given by $\alpha = 0.99$, indicating strong correlation in time with approximately 10 days decorrelation time;
5. the time interval is from 0 to 50 time units, with the same Δt as before;
6. the total number of measurements is 25 which gives the same measurement density in time as before.

In case C we will not attempt to estimate any bias term and the initial statistics for β is $\mathcal{N}(0.0, 0.0)$, i.e., zero mean and variance. The purpose is to examine how a time-correlated model error can correct for the model bias. The results from case C are shown in Fig. 5 for the EnKF (upper plot), and EnKS (lower plot) respectively. In the EnKF case it is seen how the model trend term with $\eta = 1.0$ introduces a positive drift during the ensemble integration, resulting in an overestimate at measurement times. The estimated model error has a negative trend and varies around -1, thus it partly compensates for the positive model bias. However, the nature of the stocastic forcing is to relax back towards zero between the measurement updates and it will not converge towards a fixed value. For the EnKS the situation is similar. The estimate smooths the measurements very well, and the estimated model error is varying around -1 to compensate for the positive trend term in the model. Thus, the use of time-correlated model errors may compensate for possible model biases.

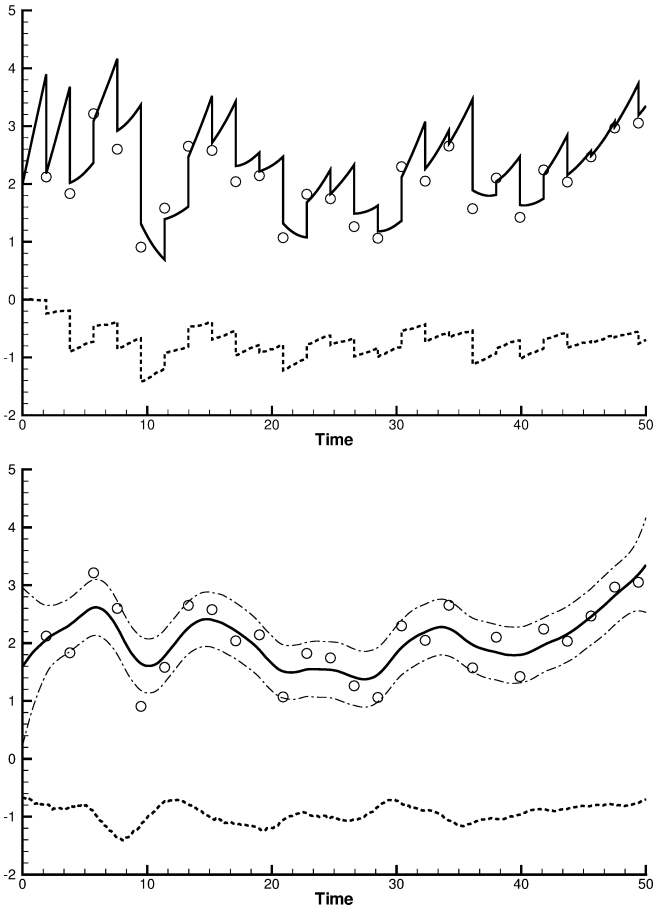

Fig. 5 Case C. The *upper plot* shows the EnKF estimate as the *thick solid line* and the estimated model error as the *dashed line*. The *lower plot* shows the EnKS estimate as the *thick solid line* while the *dotted lines* indicate the standard deviation of the estimate. The estimated model error is plotted as the *dashed line*

In case D we set the statistics for β to be $\mathcal{N}(0.0, 2.0)$, i.e., our first guess of the model bias is zero, but we assume that the error in this guess has variance equal to 2.0. The results from case D are given in Fig. 6 for the EnKF (upper plot) and EnKS (lower plot) respectively.

In the EnKF case, the estimated model bias term converges toward a value $\beta_k = -0.845$. Thus, it accounts for, and corrects, 84.5% of the model bias introduced by the trend term η. The estimate will always be located somewhere between the first guess, $\beta_0 = 0.0$, and the bias which is -1.0. A better fit to -1 can be obtained by using a larger variance on the first guess of β or by assimilating additional measurements. The model error term now varies around 0.0 possibly with a small negative trend which accounts for the uncorrected 15.5% of the bias.

In the EnKS case, the estimated bias is a constant through-out the time interval. This is obvious since there is no dynamical evolution of β_k, thus β_k is constructed from:

$$\beta_k = \beta_0 \prod_{i=1}^{25} \boldsymbol{X}_5(i), \quad \forall k. \quad (125)$$

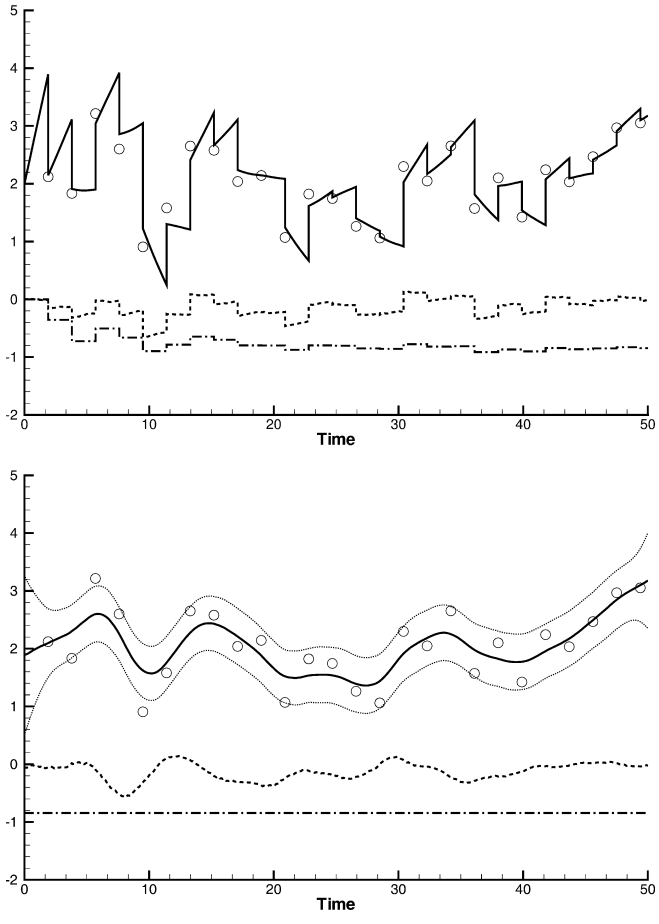

Fig. 6 Case D. Same as Fig. 5 but with the *additional dash-dot line* showing the estimated bias for the EnKF and the EnKS

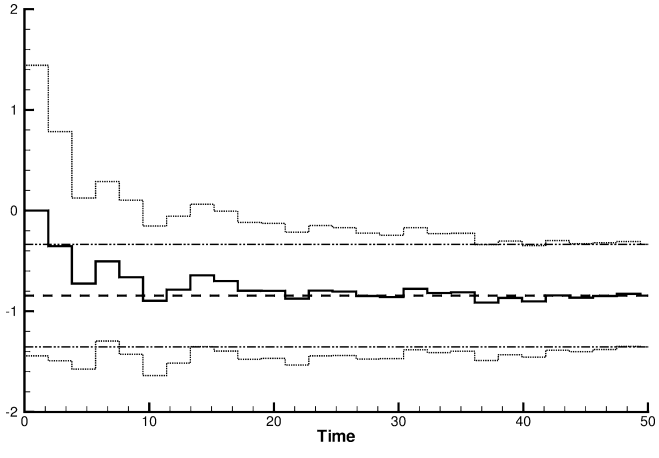

Fig. 7 Case D. The time evolution of the bias estimates from the EnKF (*solid line*) and the EnKS (*dashed line*), with the error standard deviations for the EnKF (*dotted line*) and the EnKS (*dash-dot line*)

The value for β is also equal to the final value obtained by the EnKF. Thus, there is no improvement obtained in the estimate for β by using the EnKS (see Fig. 7). Figure 7 also illustrates the convergence over time of the bias estimates and their error bars. Also in the EnKS the model error is varying around zero but with a weak negative trend. Thus, it is possible to use the EnKF and EnKS for parameter and bias estimation in dynamical models.

Acknowledgements I would like to express my thanks and gratitude to co-workers at the Nansen Center and elsewhere for providing valuable inputs and stimulating discussions during the work with this paper. In particular I would like to thank L. Bertino, K. A. Lister, Y. Morel, L. J. Natvik, D. Obaton, and H. Sagen, who have helped implementing and testing the new algorithms, checked derivations and contributed to making this paper readable and hopefully useful for the community developing ensemble methods for data assimilation. I am also grateful for comments by two anonymous reviewers, which helped me to improve the consistency of the paper. This work was supported by the EC FP–5 projects TOPAZ (EVK3-CT2000-00032) and ENACT (EVK2-CT-2001-00117), and has received support from The Research Council of Norway (Programme for Supercomputing) through a grant of computing time.

References

Allen JI, Eknes M, Evensen G (2002) An Ensemble Kalman Filter with a complex marine ecosystem model: hindcasting phytoplankton in the Cretan Sea. Annal Geophys 20: 1–13

Anderson JL (2001) An ensemble adjustment Kalman filter for data assimilation. Mon Weather Rev 129: 2884–2903

Anderson JL (2003) A local least-squares framework for ensemble filtering. Mon Weather Rev 131: 634–642

Anderson JL, Anderson SL (1999) A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. Mon Weather Rev 127: 2741–2758

Bennett AF (1992) Inverse methods in physical oceanography. Cambridge University Press, Cambridge

Bennett AF (2002) Inverse modeling of the ocean and atmosphere. Cambridge University Press, Cambridge

Bennett AF, Chua BS (1994) Open-ocean modeling as an inverse problem: the primitive equations. Mon Weather Rev 122: 1326–1336

Bennett AF, Leslie LM, Hagelberg CR, Powers PE (1993) Tropical cyclone prediction using a barotropic model initialized by a generalized inverse method. Mon Weather Rev 121: 1714–1729

Bennett AF, Chua BS, Leslie LM (1996) Generalized inversion of a global numerical weather prediction model. Meteorol Atmos Phys 60: 165–178

Bertino L, Evensen G, Wackernagel H (2002) Combining geostatistics and Kalman filtering for data assimilation in an estuarine system. Inverse Methods 18: 1–23

Bishop CH, Etherton BJ, Majumdar SJ (2001) Adaptive sampling with the ensemble transform Kalman filter. part I. Theoretical aspects. Mon Weather Rev 129: 420–436

Bleck R, Rooth C, Hu D, Smith LT (1992) Salinity-driven thermohaline transients in a wind- and thermohaline-forced isopycnic coordinate model of the North Atlantic. J Phys Oceanogr 22: 1486–1515

Brasseur P, Ballabrera J, Verron J (1999) Assimilation of altimetric data in the mid-latitude oceans using the SEEK filter with an eddy-resolving primitive equation model. J Marine Sys 22: 269–294

Brusdal K, Brankart J, Halberstadt G, Evensen G, Brasseur P, van Leeuwen PJ, Dombrowsky E, Verron J (2003) An evaluation of ensemble based assimilation methods with a layered OGCM. J Marine Sys 40-41: 253–289

Burgers G, van Leeuwen PJ, Evensen G (1998) Analysis scheme in the ensemble Kalman Filter. Mon Weather Rev 126: 1719–1724

Carmillet V, Brankart JM, Brasseur P, Drange H, Evensen G (2001) A singular evolutive extended Kalman filter to assimilate ocean color data in a coupled physical-biochemical model of the North Atlantic. Ocean Modelling 3: 167–192

Courtier P (1997) Dual formulation of variational assimilation. Q J R Meteorol Soc 123: 2449–2461
Courtier P, Talagrand O (1987) Variational assimilation of meteorological observations with the adjoint vorticity equation II: numerical results. Q J R Meteorol Soc 113: 1329–1347
Courtier P, Thepaut, Hollingsworth A (1994) A strategy for operational implementation of 4DVAR, using an incremental approach. Q J R Meteorol Soc 120: 1367–1387
Crow WT, Wood EF (2003) The assimilation of remotely sensed soil brightness temperature imagery into a land surface model using Ensemble Kalman Filtering: a case study based on ESTAR measurements during SGP97. Adv Water Resources 26: 137–149
Echevin V, Mey PD, Evensen G (2000) Horizontal and vertical structure of the representer functions for sea surface measurements in a coastal circulation model. J Phys Oceanogr 30: 2627–2635
Eknes M, Evensen G (2002) An Ensemble Kalman Filter with a 1-D marine ecosystem model. J Marine Sys 36: 75–100
Evensen G (1992) Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model. J Geophys Res 97(C11): 17 905–17 924
Evensen G (1994a) Inverse Methods and data assimilation in nonlinear ocean models. Physica (D) 77: 108–129
Evensen G (1994b) Sequential data assimilation with a non-linear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. J Geophys Res 99(C5): 10 143–10 162
Evensen G (1997) Advanced data assimilation for strongly nonlinear dynamics. Mon Weather Rev 125: 1342–1354
Evensen G, van Leeuwen PJ (1996) Assimilation of Geosat altimeter data for the Agulhas Current using the Ensemble Kalman Filter with a quasi-geostrophic model. Mon Weather Rev 124: 85–96
Evensen G, van Leeuwen PJ (2000) An Ensemble Kalman Smoother for nonlinear dynamics. Mon Weather Rev 128: 1852–1867
Evensen G, Dee D, Schröter J (1998) Parameter estimation in dynamical models. In: Chassignet EP, Verron J (eds.) Ocean modeling and parameterizations Kluwer Academic, The Nederlands, pp 373–398
Ggradshteyn IS, Ryzhik IM (1979) Table of integrals, series, and products: corrected and enlarged edition. Academic Press, New york
Grønnevik R, Evensen G (2001) Application of ensemble based techniques in fish-stock assessment. Sarsia 86: 517– 526
Hamill TM, Snyder C (2000) A hybrid Ensemble Kalman Filter–3 D variational analysis scheme. Mon Weather Rev 128: 2905–2919
Hamill TM, Mullen SL, Snyder C, Toth Z, Baumhefner DP (2000) Ensemble forecasting in the short to medium range: report from a workshop. Bull Am Meteorol Soc 81(11): 2653–2664
Hamill TM, Whitaker JS, Snyder C (2001) Distance-dependent filtering of background error covariance estimates in an Ensemble Kalman Filter. Mon Weather Rev 129: 2776–2790
Hansen JA, Smith LA (2001) Probabilistic noise reduction. Tellus, Ser (A) 53: 585–598
Haugen VE, Evensen G (2002) Assimilation of SLA and SST data into an OGCM for the Indian ocean. Ocean Dynamics 52: 133–151
Haugen VE, Evensen G, Johannessen OM (2002) Indian Ocean circulation: an integrated model and remote sensing study. J Geophys Res 107(C5): 11–1–11–23
Heemink AW, Verlaan M, Segers AJ (2001) Variance reduced ensemble Kalman Filtering. Mon Weather Rev 129: 1718– 1728
Houtekamer PL, Mitchell HL (1998) Data assimilation using an Ensemble Kalman Filter technique. Mon Weather Rev 126: 796–811
Houtekamer PL, Mitchell HL (1999) Reply. Mon Weather Rev 127: 1378–1379
Houtekamer PL, Mitchell HL (2001) A sequential ensemble Kalman filter for atmospheric data assimilation. Mon Weather Rev 129: 123–137
Jazwinski AH (1970) Stochastic processes and filtering theory. Academic, San Diego, California
Keppenne CL (2000) Data assimilation into a primitive-equation model with a parallel Ensemble Kalman Filter. Mon Weather Rev 128: 1971–1981
Keppenne CL, Rienecker M (2003) Assimilation of Temperature into an isopycnal ocean general circulation model using a parallel Ensemble Kalman Filter. J Mar Sys 40-41: 363–380
Lermusiaux PFJ (2001) Evolving the subspace of the three-dimensional ocean variability: Massachusetts Bay. J Marine Sys 29: 385–422
Lermusiaux PFJ, Robinson AR (1999a) Data assimilation via error subspace statistical estimation. part I. Theory and schemes. Mon Weather Rev 127: 1385–1407
Lermusiaux PFJ, Robinson AR (1999b) Data assimilation via error subspace statistical estimation, part II. Middle Atlantic Bight shelfbreak front simulations and ESSE validation. Mon Weather Rev 127: 1408–1432
Madsen H, Cañizares R (1999) Comparison of Extended and Ensemble Kalman filters for data assimilation in coastal area modelling. Int J Numer Meth Fluids 31: 961–981
Majumdar SJ, Bishop CH, Etherton BJ, Szunyogh I, Toth Z (2001) Can an ensemble transform Kalman filter predict the reduction in forecast-error variance produced by targeted observations? Q J R Meteorol Soc 127: 2803–2820
Miller RN, Ehret LL (2002) Ensemble generation for models of multimodal systems. Mon Weather Rev 130: 2313–2333
Miller RN, Ghil M, Gauthiez F (1994) Advanced data assimilation in strongly nonlinear dynamical systems. J Atmos Sci 51: 1037–1056
Miller RN, Carter EF, Blue ST (1999) Data assimilation into nonlinear stochastic models. Tellus Ser (A) 51: 167–194
Mitchell HL, Houtekamer PL (2000) An adaptive Ensemble Kalman Filter. Mon Weather Rev 128: 416–433
Mitchell HL, Houtekamer PL, Pellerin G (2002) Ensemble size, and model-error representation in an Ensemble Kalman Filter. Mon Weather Rev 130: 2791–2808
Natvik LJ (2001) A data assimilation system for a 3– dimensional biochemical model of the North Atlantic. PhD Thesis, University of Bergen/Nansen Environmental and Remote Sensing Center, Edv. Griegsv 3a, 5059 Bergen, Norway
Natvik LJ, Evensen G (2003a) Assimilation of ocean colour data into a biochemical model of the North Atlantic, part 1. Data assimilation experiments. J Marine Sys 40-41: 127–153
Natvik LJ, Evensen G (2003b) Assimilation of ocean colour data into a biochemical model of the North Atlantic, part 2. Statistical analysis. J Marine Sys 40-41: 155–169
Park JH, Kaneko A (2000) Assimilation of coastal acoustic tomography data into a barotropic ocean model. Geophys Res Lett 27: 3373–3376
Pham DT (2001) Stochastic methods for sequential data assimilation in strongly nonlinear systems. Mon Weather Rev 129: 1194–1207
Pham DT, Verron J, Roubaud MC (1998) A singular evolutive extended Kalman Filter for data assimilation in oceanography. J Marine Sys 16: 323–340
Reichle RH, McLaughlin DB, Entekhabi D (2002) Hydrologic data assimilation with the Ensemble Kalman Filter. Mon Weather Rev 130: 103–114
Talagrand O, Courtier P (1987) Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. Q J R Meteorol Soc 113: 1311–1328
Thacker WC, Esenkov OE (2002) Assimilating XBT data into HYCOM. J Atmos Ocean Tech 19: 709–724
Tippett MK, Anderson JL, Bishop CH, Hamill TM, Whitaker JS (2003) Ensemble square-root filters. Mon Weather Rev 131: 1485–1490
Troccoli A, Balmaseda MA, Segschneider J, Vialard J, Anderson DLT (2002) Salinity adjustment in the presence of temperature data assimilation. Mon Weather Rev 130: 89–102
van Leeuwen PJ (1999a) Comment on ''Data assimilation using an Ensemble Kalman Filter technique''. Mon Weather Rev 127: 6

van Leeuwen PJ (1999b) The time mean circulation in the Agulhas region determined with the Ensemble Smoother. J Geophys Res 104: 1393–1404
van Leeuwen PJ (2001) An Ensemble Smoother with error estimates. Mon Weather Rev 129: 709–728
van Leeuwen PJ (2003) A variance-minimizing filter for large-scale applications. Mon Weather Rev (In press)
van Leeuwen PJ, Evensen G (1996) Data assimilation and inverse methods in terms of a probabilistic formulation. Mon Weather Rev 124: 2898–2913
van Loon M, Builtjes PJH, Segers AJ (2000) data assimilation of ozone in the atmoshperic transport chemistry model LOTUS. Environ Modelling Software 15: 603–609
Verlaan M, Heemink AW (2001) Nonlinearity in data assimilation applications: a practical method for analysis. Mon Weather Rev 129: 1578–1589
Whitaker JS, Hamill TM (2002) Ensemble data assimilation without perturbed observations. Mon Weather Rev 130: 1913–1924