Tellus **000**, 000–000 (0000) Printed 12 July 2008 (Tellus LaTeX style file v2.2)

EnVE: A consistent hybrid ensemble/variational estimation strategy for multiscale uncertain systems.

T. Bewley, J. Cessna,* and C. Colburn
Flow Control Lab, Dept. of MAE, UC San Diego, La Jolla, CA 92093, USA

(Manuscript received 1 January 2008; in final form 1 January 2008)

ABSTRACT

Chaotic systems are characterized by long-term unpredictability. Existing methods designed to estimate and forecast such systems, such as Extended Kalman filtering (a "sequential" or "incremental" matrix-based approach) and 4DVar (a "variational" or "batch" vector-based approach), are essentially based on the assumption that Gaussian uncertainty in the initial state, state disturbances, and measurement noise lead to uncertainty of the state estimate at later times that is well described by a Gaussian model. This assumption is not valid in chaotic systems with appreciable uncertainties. A new method is thus proposed that combines the speed and LQG optimality of a sequential-based method, the non-Gaussian uncertainty propagation of an ensemble-based method, and the favorable smoothing properties of a variational-based method. This new approach, referred to as Ensemble Variational Estimation (EnVE), is a natural extension of the Ensemble Kalman and 4DVar algorithms. EnVE is a hybrid method leveraging sequential preconditioning of the batch optimization steps, simultaneous backward-in-time marches of the system and its adjoint (eliminating the checkpointing normally required by 4DVar), a receding-horizon optimization framework, and adaptation of the optimization horizon based on the estimate uncertainty at each iteration. If the system is linear, EnVE is consistent with the well-known Kalman filter, with all of its well-established optimality properties. The strength of EnVE is its remarkable effectiveness in highly uncertain nonlinear systems, in which EnVE consistently uses and *revisits* the information contained in recent observations with batch (that is, variational) optimization steps, while consistently propagating the uncertainty of the resulting estimate forward in time.

1 Introduction

The estimation and forecasting of chaotic, multiscale, uncertain fluid systems is one of the most highly visible computational grand challenge problems of our generation. Specifically, this class of problems includes weather forecasting, climate forecasting, and flow control. The financial impact of a hurricane passing through a major metropolitan center regularly exceeds a billion dollars. Improved forecasting techniques provide early and accurate warnings, which are critical to minimize the impact of such events. On longer time scales, the estimation and forecasting of changes in ocean currents and temperatures is essential for an improved understanding of changes to the earth's weather systems. On shorter time scales, feedback control of fluid systems (for reasons such as minimizing drag, maximizing harvested energy, etc.) in mechanical, aerospace, environmental, and chemical engineering settings lead to a variety of similar estimation problems. While this paper makes no claims with regards to solving such important problems, it does introduce a new hybrid ensemble/variational strategy for the estimation and forecasting of such multiscale uncertain fluid systems that might well have a transformational effect in all of these areas.

Much of the research today in data assimilation for multiscale uncertain fluid systems is focused on medium to short-range weather forecasting. To this end, the field of data assimilation has matured greatly in the past two decades. First, with the development of spatial (three-dimensional) variational data assimilation (3DVar)–see, e.g., Parrish & Derber (1992) and Lorenc (1986)–a consistent statistical framework was formed that could be utilized for large-scale atmospheric systems. This was followed by a type of spatial/temporal (four-dimensional) variational data assimilation (4DVar)–see, e.g., Le Dimet & Talagrand (1986) and Rabier et al. (1998)–in which the consistent statistical framework was extended to include a time history of observations. It has been shown by Li & Navon (2001) that this spatial/temporal framework has the effect of conditioning the resulting estimate on all included data, as does the Kalman Smoother [see Rauch et al. (1965) and Cohn et al. (1994)].

4DVar was developed in parallel, and largely independently, in the controls and weather forecasting communities. In the controls community, the technique is referred to as Moving Horizon Estimation (MHE), as discussed in Michalska & Mayne (1995). MHE was developed with low dimensional ODE systems in mind; implementations of MHE typically search for a small time-varying "state disturbance" or "model error" term in addition to the initial state of the system in order reconcile the measurements with the model over the period of interest as accurately as possible. 4DVar, in contrast, was developed with high dimensional discretizations of infinite-dimensional (PDE) systems in mind; in order to retain numerical tractability, implementations of 4DVar typically do not search for such a time-varying model error term.

Another technique that has been introduced to accelerate MHE/4DVar implementations is multiple shooting–see, e.g., Kraus et al. (2006). With this technique, the horizon of interest is split into two or more subintervals. The initial conditions (and, in some im-

* Corresponding author.
e-mail: jcessna@ucsd.edu

plementations, the time-varying model error term) for each subinterval are first initialized and optimized independently, then these several independent solutions are adjusted so that the trajectories coincide at the matching points between the subintervals.

The traditional Kalman [see Kalman (1960) and Kalman & Bucy (1961)] and extended Kalman filtering ideas were explored by Ghil et al. (1981) for atmospheric applications. These methods require the computation of a reduced-rank approximation of the covariance matrix at the heart of the Kalman filter in order to be tractable in high-dimensional systems. Such an approximation is now known as Chandresarkhar's method, and was introduced by Kailath (1973).

The more recent development of the Ensemble Kalman Filter (EnKF) [see, e.g., Evensen (1994), Houtekamer & Mitchell (1998), Houtekamer & Mitchell (2001), Evensen (2003), and the references contained therein] has focused much attention on an important refinement of this sequential method in which the estimation statistics are intrinsically represented via the distribution of a cluster or "ensemble" of state estimates in phase space. The simultaneous simulation of several perturbed trajectories of the state estimate eliminates the need to propagate the entire state covariance matrix along with the estimate as required by traditional Kalman and extended Kalman approaches. Instead, this covariance information is approximated based on the spread of the ensemble members in order to compute a Kalman-like sequential update at the measurement times (for further discussion, see Section 2.2).

Since its introduction, the EnKF has spawned many variations and modifications that seek to improve both its performance and its numerical tractability. For example, Kalman square-root filters update the analysis only once, in a manner different than the traditional perturbed observation method. Some square-root filters introduced include the ensemble adjustment filter by Anderson (2001), the ensemble transform filter by Bishop et al. (2001), and the ensemble square-root filter by Whitaker & Hamill (2002). Work has also been done by, e.g., Kim et al. (2003), to further relax the linear Gaussian assumptions with regards to the interpolation between the observation and the background statistics. The unscented Kalman filter, first introduced by Wan & van der Merwe (2000), derives a more accurate particle propagation of the estimate covariance, but requires far too many ensemble members to remain tractable for multiscale systems. Another essential advancement in the implementation of the EnKF is the idea of covariance localization, as discussed in Hamill et al. (2001) and Ott et al. (2004). With covariance localization, spurious correlations of the uncertainty covariance over large distances are reduced in an ad hoc fashion in order to improve the overall performance of the estimation algorithm. This adjustment is motivated by the rank-deficiency of the ensemble approximation of the covariance matrix, and facilitates parallel implementation of the resulting algorithm.

For nonlinear systems, the EnKF framework is suboptimal due to its reliance on a Kalman-like measurement update formula. This update formula is, effectively, based on a Gaussian distribution of the estimate uncertainty. The more general Particle Filter (PF) propagates a set of "particles" representing several potential trajectories of the system in a very similar manner as the EnKF propogates its ensemble. In the PF method, however, each particle has an associated "weighting factor" that is used to compute a biased mean and corresponding higher moment statistics. Unlike the EnKF, at the measurement times, the particle filter uses the new observations to update the weighting factor of each particle, without actually updating the particle's position in phase space. As a result, in the limit of an infinite number of particles, this update strategy can be shown to be optimal, even for nonlinear systems with non-Gaussian uncertainties. Unfortunately, compared to the EnKF, the PF method requires excessive of computational resources in multiscale systems due to the relatively large number of particles required for adequate performance. Further, particle re-population strategies which "prune" particles with low weights from the set, and then initialize new particles near the current best estimate, are computationally intensive. Nevertheless, the Particle Kalman Filter (PKF) method proposed by Hoteit et al. (2008), which attempts to combine the PF and EnKF approaches in order to inherit the non-Gaussian uncertainty characterization of the PF method and the numerical tractability of the EnKF method, appears to be quite promising; this method could potentially benefit directly from a further hybridization with the variational approach, as propsed here.

The two modern schools of thought in data assimilation for multiscale uncertain systems (namely, 4DVar and EnKF) have, for the most part, remained largely independent, despite their similar theoretical backgrounds. The data assimilation community today has made considerable efforts to compare and contrast both the performance and the theoretical foundation of these two methods [see, e.g., Lorenc (2003), Caya et al. (2005), Kalnay et al. (2007), and Gustaffson (2007)]. While these comparisons are enlightening, it is quite possible that the optimal data assimilation solution for many cases may well be a *hybrid* combination of the two methods, as suggested by Gustaffson (2007). We have identified five recent attempts at such hybridization:

(i) the 3DVar/EnKF method of Hamill & Snyder (2000),
(ii) the EnKS method of Evensen & van Leeuwen (2000),
(iii) the 4DEnKF method of Hunt et al. (2004),
(iv) the VAE method of Berre et al. (2007), and
(v) the E4DVAR method of Zhang et al. (2007).

The 3DVar/EnKF algorithm introduced by Hamill & Snyder (2000) utilizes the ensemble framework to propagate the estimate statistics in a nonlinear setting, but does not exploit the temporal smoothing characteristics of the 4DVar algorithm. The EnKS (Ensemble Kalman Smoother) method developed by Evensen & van Leeuwen (2000) recomputes a new analysis, essentially from scratch, for all recent measurements upon the receipt of each new observation; this approach is computationally intractable for multiscale systems. The 4DEnKF (4D Ensemble Kalman Filter) introduced by Hunt et al. (2004) provides a means for assimilating past (and non-uniform) observations in a sequential framework, but does not intrinsically smooth the resulting estimate or fully implement the 4DVar framework. The VAE (Variational Assimilation Ensemble) method of Berre et al. (2007) runs a half dozen perturbed decoupled 4DVar or 3DFgat[1] assimilations in parallel to estimate error covariances, but does not fundamentally integrate the EnKF and 4Dvar concepts to obtain a hybrid method. The E4DVAR (Ensemble 4DVar) method discussed by Zhang et al. (2007), which is the closest existing method to that proposed here, runs a 4DVar and EnKF in parallel, sequentially shifting the mean of the ensemble based on the 4DVar result and providing the background term of the 4DVar algorithm based on the EnKF result; however, this method does not attempt a tighter coupling of the EnKF and 4DVar approaches by using an Ensemble Smoother to initialize better (and, thus, accelerate) the variational iteration.

[1] That is, 3D First Guess at the Apprpriate Time (3DFgat), an intermediate variational method with complexity somewhere between that of 3DVar and 4DVar [see Fisher (2002)].

The proposed new algorithm, Ensemble Variational Estimation (EnVE), is, a consistent and tightly-coupled hybrid of the traditional sequential (EnKF) and variational (4DVar/MHE) methods. EnVE uses the statistical properties of a sequential ensemble Kalman smoother (EnKS) to, from time to time, precondition a variational assimilation step. In the earlier work done by Cessna et al. (2007), the 4DVar/MHE framework was inverted, promoting "retrograde" time marches (that is, marching the state estimate *backward* in time and the corresponding adjoint *forward* in time), which facilitates an *adaptive* (i.e., multiscale-in-time) receding-horizon optimization framework. The motivation behind this original work was sound, but the algorithm lacked the consistency necessary to account for the background estimate statistics. With the incorporation of the EnKF, creating the new EnVE algorithm proposed here, it is possible to retain the adjustable optimization horizons facilitated by this retrograde setting while simultaneously eliminating the typical storage problem associated with variational methods. Special significance is placed in this paper on the *consistency* of EnVE; specifically, that the algorithm converges to the optimal Kalman filter solution in the LQG setting.

Section 2 of this paper reviews briefly the general forms of both the EnKF and 4DVar. The adjoint for a continuous-time model with discrete-time measurements is fully derived, as most existing derivations deal with either the fully continuous [see, e.g., Kim & Bewley (2007)] or fully discrete [see, e.g., Bouttier & Courtier (2002)] formulations. Section 3 outlines the theoretical foundations of the EnVE algorithm, and derives (apparently, for the first time) the backward-in-time Kalman filter "downdate" equation, which exactly inverts the classical discrete-time Kalman filter update equation in a numerically tractable manner. Some numerical considerations (with regards to implementation of EnVE in an MPI setting) are then described in Section 4. The importance of consistency, and how it relates to the EnVE algorithm, is further clarified in Section 5. The primary advantages of the EnVE formulation are sumarized in Section 6. The full EnVE algorithm is demonstrated on a simple example of chaos, the Lorenz system, in Section 7. Two follow-up papers [see Bewley et al. (2008a, 2008b)] detail the implementation of the EnVE algorithm on 1D, 2D, and 3D chaotic PDE systems, and introduce a unique adaptive observation algorithm which builds directly upon the hybrid framework of the EnVE algorithm.

2 Background

Ensemble Variational Estimation (EnVE) is a consistent hybrid data assimilation method that combines the key ideas of the sequential Ensemble Kalman Filter (EnKF) method and the batch (in time) variational method known as 4DVar in the weather forecasting community and as Moving Horizon Estimation (MHE) in the controls community. Thus, these methods are first briefly reviewed independently. Without loss of generality, the dynamic model used to introduce these methods is a continuous-time nonlinear ODE system with discrete-time measurements:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t),\mathbf{w}(t)), \tag{1a}$$

$$\mathbf{y}_k = h(\mathbf{x}(t_k)) + \mathbf{v}_k, \tag{1b}$$

where the state disturbance $\mathbf{w}(t)$ is a zero-mean, continuous-time random process with autocorrelation

$$R_{\mathbf{w}}(\tau;t) = E\{\,\mathbf{w}(t+\tau)\,\mathbf{w}^H(t)\,\} = Q\delta^\sigma(\tau), \tag{2a}$$

$$\text{where} \qquad \delta^\sigma(\tau) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\tau^2/(2\sigma^2)}, \tag{2b}$$

with $Q \geqslant 0$ and[2] $0 < \sigma \ll 1$, and the measurement noise $\mathbf{v}_k$ is a zero-mean, white, discrete-time random process with autocorrelation

$$R_{\mathbf{v}}(j;k) = E\{\,\mathbf{v}_{k+j}\,\mathbf{v}_k^H\,\} = \mathcal{R}\delta_{j0}, \tag{3}$$

with $\mathcal{R} > 0$. Is also assumed that $\mathbf{w}(t)$ and $\mathbf{v}_k$ are uncorrelated.

The noisy measurements $\mathbf{y}_k$ are assumed to be taken at time $t_k = k\Delta t$ for a fixed sample period Δt. For the purposes of analysis, these observations are assumed available for a long history into the past, up to and including the present time of the system being estimated, which is often renormalized to be $t = t_0 = 0$. It is useful to think of the present time as the time of the most recent available measurement, so, accordingly, this measurement will be denoted $\mathbf{y}_0$ at the beginning of each analysis step. This sets the basis for the indexing notation used in this paper: $k = 0$ represents the index of the most recent measurement, $k \leqslant 0$ is the set of indices of all available measurements, and $k > 0$ indexes observations that are yet to be taken. Continuous-time trajectories, such as $\mathbf{x}(t)$ (the "truth" model), are defined for all time, but are frequently referenced at the observation times only. Hence, the following notation is used:

$$\mathbf{x}(k\Delta t) = \mathbf{x}(t_k) = \mathbf{x}_k. \tag{4}$$

2.1 Uncertainty Propagation in Chaotic Systems

Estimation, in general, involves the determination of a probability distribution. This probability distribution describes the likelihood that any particular point in phase space matches the truth model. That is, without knowing the actual state of a system, estimation strategies attempt to represent the probability of any given state using only a time history of noisy observations of a subset of the system and an approximate dynamic model of the system of interest. Given this statistical distribution, estimates can be inferred about the "most likely" state of the system, and how much confidence should be placed in that estimate. Unfortunately, in this most general form, the estimation problem is intractable in most systems. However, given certain justifiable assumptions about the nature of the model and its associated disturbances, simplifications can be applied with regards to how the probability distributions are modeled. Specifically, in linear systems with Gaussian uncertainty of the initial state, Gaussian state disturbances, and Gaussian measurement noise, it can be shown that the probability distribution of the optimal estimate is itself Gaussian [see, e.g., Anderson & Moore (1979)]. Consequently, the entire distribution of the estimate in phase space can be represented exactly by its mean $\overline{\mathbf{x}}$ and its second moment about the mean (that is, its covariance), $\mathcal{P}$, where

$$\mathcal{P} = E\left[(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^H\right]. \tag{5}$$

This is the essential piece of theory that leads to the traditional Kalman Filter (KF), first introduced by Kalman (1960) and Kalman & Bucy (1961).

Sequential data assimilation methods provide a method to

[2] The case for infinitesimal σ is sometimes referred to as "continuous-time white noise", but presents certain technical difficulties [Bewley (2008)].

propagate the mean $\bar{\mathbf{x}}$ and covariance $\mathcal{P}$ forward in time, making the appropriate updates to both upon the receipt of each new measurement. Under the assumption of a linear system and white Gaussian state disturbances and measurement noise, the uncertainty distribution of the optimal estimate is itself Gaussian, and thus is *completely* described by the mean estimate $\bar{\mathbf{x}}$ and the covariance $\mathcal{P}$ propagated by the Kalman formulation. It is useful to think of these quantities, at any given time t_k, as being conditioned on a subset of the available measurements. The notation $\bar{\mathbf{x}}_{k|j}$ represents the highest likelihood estimate at time t_k given measurements up to and including time t_j. Similarly, $\mathcal{P}_{k|j}$ represents the corresponding covariance of this estimate. In particular, $\bar{\mathbf{x}}_{k|k-1}$ and $\mathcal{P}_{k|k-1}$ are often called the prediction estimate and prediction covariance, whereas $\bar{\mathbf{x}}_{k|k}$ and $\mathcal{P}_{k|k}$ are often called the current estimate and the current covariance. Note that $\bar{\mathbf{x}}_{k|k+K}$, for some $K > 0$, is often called a smoothed estimate, and may be obtained in the sequential setting by a Kalman smoother [see, Rauch et al. (1965) and Anderson & Moore (1979)].

For nonlinear systems with relatively small uncertainties, a common variation on the KF known as the Extended Kalman Filter (EKF) has been developed in which the mean and covariance are propagated, to first-order accuracy, about a linearized trajectory of the full system. Essentially, if a Taylor-series expansion for the nonlinear evolution of the covariance is considered, and all terms higher than quadratic are dropped, what is left is the differential Riccati equation associated with the EKF covariance propagation. Though this approach gives acceptable estimation performance for nonlinear systems when uncertainties are small as compared to the fluctuations of the state itself, EKF estimators often diverge when uncertainties are more substantial, and other techniques are needed.

At its core, the linear thinking associated with the uncertainty propagation in the KF and EKF breaks down in chaotic systems. Chaotic systems are characterized by stable manifolds or "attractors" in n-dimensional phase space. Such attractors are fractional-dimensional subsets (a.k.a. "fractal" subsets) of the entire phase-space. Trajectories of chaotic systems are stable with respect to the attractor in the sense that initial conditions off the attractor converge exponentially to the attractor, and trajectories on the attractor remain on the attractor. On the attractor, however, trajectories of chaotic systems are characterized by an *exponential divergence*–along the attractor–of slightly perturbed trajectories. That is, two points infinitesimally close on the attractor at one time will diverge exponentially from one another as the system evolves until they are effectively uncorrelated.

Just as an individual trajectories diverge along the attractor, so does the uncertainty associated with them. This uncertainty diverges in a highly non-Gaussian fashion when such uncertainties are not infinitesimal (see Figure 1). Estimation techniques that attempt to propagate probability distributions under linear, Gaussian assumptions fail to capture the true uncertainty of the estimate in such settings, and thus improved estimation techniques are required. The Ensemble Kalman Filter, in contrast, accounts properly for the nonlinearities of the chaotic system when propagating estimator uncertainty. This idea is a central component of the hybrid ensemble/variational method proposed in the present work, and is thus reviewed next.

2.2 *Ensemble Kalman Filtering*

The Ensemble Kalman Filter (EnKF) is a sequential data assimilation method useful for nonlinear multiscale systems with substantial uncertainties. In practice, it has been shown repeatedly to provide significantly improved state estimates in systems for

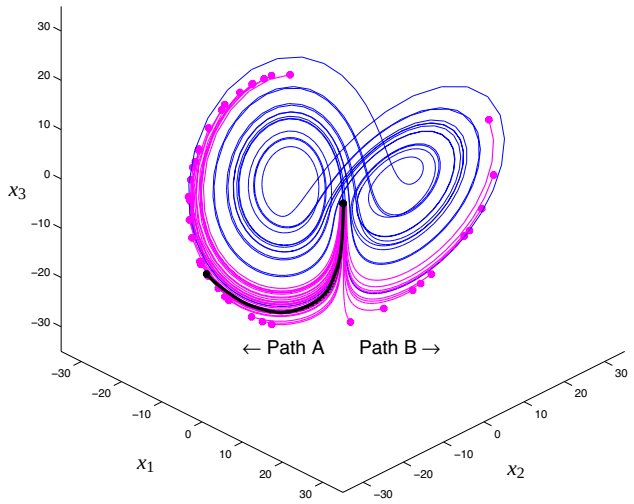

Figure 1. Non-Gaussian uncertainty propagation in the Lorenz system. The black point in the center shows a typical point located in a sensitive area of this chaotic system's attractor in phase space, representing a current estimate of the state. The thick black line represents the evolution in time of the trajectory from this estimate. If the uncertainty of the estimate is modeled as a very small cloud of points, centered at the original estimate with an initially Gaussian distribution, then the additional magenta lines show the evolution of each of these perturbed points in time. A Gaussian model of the resulting distribution of points is, clearly, completely invalid.

which the traditional EKF breaks down. Unlike in the KF and EKF, the statistics of the estimation error in the EnKF are not propagated via a covariance matrix, but rather are implicitly approximated via the appropriate nonlinear propagation of several perturbed trajectories ("ensemble members") centered about the ensemble mean, as illustrated in Figure 1. The collection of these ensemble members (itself called the "ensemble"), propagates the statistics of the estimation error exactly in the limit of an infinite number of ensemble members. Realistic approximations arise when the number of ensemble members, N, is (necessarily) finite. Even with a finite ensemble, the propagation of the statistics is still consistent with the nonlinear nature of the model. Conversely, the EKF propagates only the lowest-order components of the second-moment statistics about some assumed trajectory of the system. This difference is a primary strength of the EnKF.

In practice, the ensemble members $\hat{\mathbf{x}}^j$ in the EnKF are initialized with some known statistics about an initial mean estimate $\bar{\mathbf{x}}$. The ensemble members are propagated forward in time using the fully nonlinear model equation (1a), incorporating random forcing $\mathbf{w}^j(t)$ with statistics consistent with those of the actual state disturbances $\mathbf{w}(t)$ [see (2)]:

$$\frac{d\hat{\mathbf{x}}^j(t)}{dt} = f(\hat{\mathbf{x}}^j(t), \mathbf{w}^j(t)). \tag{6}$$

At the time t_k (for integer k), an observation $\mathbf{y}_k$ is taken [see (1b)]. Each ensemble member is updated using this observation, incorporating random forcing $\mathbf{v}_k^j$ with statistics consistent with those of the actual measurement noise, $\mathbf{v}_k$ [see (3)]:

$$\mathbf{d}_k^j = \mathbf{y}_k + \mathbf{v}_k^j. \tag{7}$$

Given this perturbed observation $\mathbf{d}_k^j$, each ensemble member is up-

dated in a manner consistent[3] with the KF and EKF:

$$\hat{\mathbf{x}}^j_{k|k} = \hat{\mathbf{x}}^j_{k|k-1} + \mathcal{P}^e_{k|k-1} H^H (H \mathcal{P}^e_{k|k-1} H^H + \mathcal{R})^{-1} (\mathbf{d}^j_k - H \hat{\mathbf{x}}^j_{k|k-1}), \quad (8)$$

where H is the linearization of the output operator $h(\cdot)$ in (1b). Unlike the EKF, in which the entire covariance matrix $\mathcal{P}$ is propagated using the appropriate Riccati equation, the EnKF estimate covariance $\mathcal{P}^e$ is computed "on the fly" using the second moment of the ensembles from the ensemble mean:

$$\mathcal{P}^e = \frac{(\delta\widehat{X})\,(\delta\widehat{X})^H}{N-1}, \text{ where } \delta\widehat{X} = \begin{bmatrix} \delta\hat{\mathbf{x}}^1 & \delta\hat{\mathbf{x}}^2 & \cdots & \delta\hat{\mathbf{x}}^N \end{bmatrix},$$
$$\delta\hat{\mathbf{x}}^j = \hat{\mathbf{x}}^j - \bar{\mathbf{x}}, \quad \text{and} \quad \bar{\mathbf{x}} = \frac{1}{N}\sum_j \hat{\mathbf{x}}^j, \quad (9)$$

where N is the number of ensemble members, and the time subscripts have been dropped for notational clarity[4].

Thus, like the KF and EKF, the EnKF is propagated with a forecast step (6) and an update step (8). The ensemble members $\hat{\mathbf{x}}^j(t)$ are propagated forward in time using the system equations [with state disturbances $\mathbf{w}^j(t)$] until a new measurement $\mathbf{y}_k$ is obtained, then each ensemble member $\hat{\mathbf{x}}^j(t_k) = \hat{\mathbf{x}}^j_k$ is updated to include this new information [with measurement noise $\mathbf{v}^j_k$]. The covariance matrix is not propagated explicitly, as its evolution is implicitly represented by the evolution of the ensemble itself.

It is convenient to think of the various estimates during such a data assimilation procedure in terms of the set of measurements that have been included to obtain that estimate. Just as it is possible to propagate the ensemble members forward in time accounting for new measurements, ensemble members can also be propagated *backward* in time, either retaining the effect of each measurements or subtracting this information back off. In the case of a linear system, the former approach is equivalent to the Kalman smoother, while the later approach simply retraces the forward march of the Kalman filter backward in time. In order to make this distinction clear, the notation $\widehat{X}_{j|k}$ will represent the estimate ensemble at time t_j given measurements up to and including time t_k. Similarly, $\bar{\mathbf{x}}_{j|k}$ will represent the corresponding ensemble mean; that is, the average of the ensemble and the "highest-likelihood" estimate of the system.

While the EnKF significantly outperforms the more traditional EKF for chaotic systems, further approximations need to be made for multiscale systems such as atmospheric models. When assimilating data for 3D PDEs, the discretized state dimension n is many orders of magnitude larger than the number of ensemble members N that is computationally feasible (i.e., $N \ll n$). The consequences of this are twofold. First, the ensemble covariance matrix $\mathcal{P}^e$ is guaranteed to be singular, which can lead to difficulty when trying to solve linear systems constructed with this matrix. Second, this singularity combined with an insufficient statistical sample size produces directions in phase space in which no information is gained through the assimilation. This leads to spurious correlations in the covariance that would cause improper updates across the domain of the system. This problem can be significantly diminished via the ad hoc method of "covariance localization" mentioned previously, which artificially suppresses these spurious correlations using a distance-dependent damping function.

2.3 *Variational Methods*

For high-dimensional systems in which matrix-based methods are computationally infeasible, vector-based variational methods are preferred for data assimilation. 3DVar is a vector-based equivalent to the KF. In both 3DVar and KF, the cost function being minimized is a (quadratic) weighted combination of the uncertainty in the background term and the uncertainty in the new measurement. If the system is linear, the optimal update to the state estimate can be found analytically, though this solution requires matrix-based arithmetic (specifically, the propagation of a Riccati equation), and is the origin of the optimal update gain matrix for the KF. When this matrix is too large for direct computation, a local gradient can instead be found using vector-based arithmetic only; 3DVar uses this local gradient information to determine the optimal update iteratively.

Similarly, 4DVar is the vector-based equivalent to the Kalman Smoother. In 4DVar, a finite time window (or "batch process") of a history of measurements is analyzed together to improve the estimate of the system at one edge of this window (and, thus, the corresponding trajectory of the estimate over the entire window). Unlike sequential methods, a smoother uses all available data over this finite time window to optimize the estimates of the system. This has the consequence of refining past estimates of the system based on future measurements, whereas with sequential methods any given estimate is only conditioned on the previous observations.

For analysis, let the variational window be defined as $t \in [-T, 0]$. Additionally, let there be $K+1$ measurements in this interval, with measurement indices given by the set

$$M = \{\, k \mid t_k \in [-T, 0] \,\} \;\Rightarrow\; M = \{\, -K,\ \cdots, -1,\ 0 \,\}. \quad (10)$$

Without loss of generality, it will be assumed that there are measurements at both edges of the window (i.e. $t_{-K} = -T$ and $t_0 = 0$). Then, the cost function $\mathcal{J}(\mathbf{u})$ that 4DVar attempts to minimize (with respect to $\mathbf{u}$) is defined as follows:

$$\mathcal{J}(\mathbf{u}) = \frac{1}{2}(\mathbf{u} - \bar{\mathbf{x}}_{-K|-K})^H \mathcal{P}^{-1}_{-K|-K} (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K}) +$$
$$\frac{1}{2}\sum_{k=-K}^{0} \left(\mathbf{y}_k - H\tilde{\mathbf{x}}_k\right)^H \mathcal{R}^{-1} \left(\mathbf{y}_k - H\tilde{\mathbf{x}}_k\right), \quad (11)$$

where the "optimization variable" $\mathbf{u}$ is the initial condition on the refined state estimate $\tilde{\mathbf{x}}$ on the interval $[-T, 0]$; that is,

$$\frac{d\tilde{\mathbf{x}}(t)}{dt} = f(\tilde{\mathbf{x}}(t), 0), \quad (12a)$$
$$\tilde{\mathbf{x}}_{-K} = \mathbf{u}. \quad (12b)$$

The first term in the cost function (11), known as the "background" term, summarizes the fit of $\mathbf{u}$ with the current probability distribution before the optimization (i.e., the effect of all past measurement updates). Like with the KF, $\bar{\mathbf{x}}_{-K|-K}$ is the estimate at time t_{-K} not including any of the new measurements in the window, and the covariance $\mathcal{P}^{-1}_{-K|-K}$ quantifies the second moment of the uncertainty in that estimate. Assuming an a priori Gaussian probability distribution of this uncertainty, the background mean and covariance exactly describe this distribution. The second term in the cost function (11) summarizes the misfit between the estimated system trajectory and the observations within the variational window. Thus,

[3] Note that some authors [see, e.g., Evensen (2003)] prefer to replace $\mathcal{R}$ in (8) with $\mathcal{R}^e$, where

$$\mathcal{R}^e = \frac{(V_k)\,(V_k)^H}{N-1} \quad \text{and} \quad V_k = \begin{bmatrix} \mathbf{v}^1_k & \mathbf{v}^2_k & \cdots & \mathbf{v}^N_k \end{bmatrix}.$$

Our current research has not revealed any clear advantage for using this more computationally expensive form.

[4] Note also that the factor $N-1$ (instead of N) is used in (9) to obtain an unbiased estimate of the covariance matrix [see Bewley (2008)].

the solution $\mathbf{u}$ to this optimization problem is the estimate that best "fits" the observations over the variational window while also accounting for the existing information from observations prior to the variational window.

In practice, a 4DVar iteration is usually initialized with the background mean, $\mathbf{u} = \bar{\mathbf{x}}_{-K|-K}$. Given this initial guess for $\mathbf{u}$, the trajectory $\tilde{\mathbf{x}}(t)$ may be found using the full nonlinear equations for the system (12). To find the gradient of the cost function (11), consider a small perturbation of the optimization variable, $\mathbf{u} \leftarrow \mathbf{u} + \mathbf{u}'$, and the resulting perturbed trajectory, $\tilde{\mathbf{x}}(t) \leftarrow \tilde{\mathbf{x}}(t) + \tilde{\mathbf{x}}'(t)$, and perturbed cost function, $\mathcal{J}(\mathbf{u}) \leftarrow \mathcal{J}(\mathbf{u}) + \mathcal{J}'(\mathbf{u}')$. The local gradient of (11), $\bigtriangledown\mathcal{J}(\mathbf{u})$, is defined here as the sensitivity of the perturbed cost function $\mathcal{J}'(\mathbf{u}')$ to the perturbed optimization variable $\mathbf{u}'$:

$$\mathcal{J}'(\mathbf{u}') = \big[\bigtriangledown\mathcal{J}(\mathbf{u})\big]^H \mathbf{u}'. \tag{13}$$

The following derivation illustrates how to write $\mathcal{J}'(\mathbf{u}')$ in this simple form, leveraging the definition of an appropriate adjoint field.

The full derivation of the gradient $\bigtriangledown\mathcal{J}(\mathbf{u})$ is included here due to the unusual setting considered (that is, of a continuous-time system with discrete-time measurements). Perturbing the nonlinear model equations (1a) and linearizing about $\tilde{\mathbf{x}}(t)$ gives:

$$\frac{d\tilde{\mathbf{x}}'(t)}{dt} = A\tilde{\mathbf{x}}'(t) \quad \text{with} \quad \tilde{\mathbf{x}}'_{-K} = \mathbf{u}' \tag{14}$$

$$\Rightarrow \quad \mathcal{L}\tilde{\mathbf{x}}' = 0 \quad \text{where} \quad \mathcal{L} = \frac{d}{dt} - A. \tag{15}$$

Similarly, the perturbed cost function is:

$$\mathcal{J}'(\mathbf{u}') = (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K})^H \mathcal{P}^{-1}_{-K|-K}\, \mathbf{u}' - \sum_{k=-K}^{0} (\mathbf{y}_k - H\tilde{\mathbf{x}}_k)^H \mathcal{R}^{-1} H\tilde{\mathbf{x}}'_k. \tag{16}$$

The perturbed cost function (16) is not quite in the form necessary to extract the gradient, as illustrated in (13). However, there is an implicitly defined linear relationship between $\mathbf{u}'$ and $\tilde{\mathbf{x}}'(t)$ on $t \in [-T, 0]$ given by (14). To re-express this relationship, a set of K adjoint functions $\mathbf{r}^{(k)}(t)$ are defined over the measurement intervals such that, for all $k \in [1, K]$, the adjoint function $\mathbf{r}^{(k)}(t)$ is defined on the closed interval $t \in \big[t_{-k}, t_{1-k}\big]$. These adjoint functions will be used to identify the gradient. To this end, a suitable duality pairing is defined here as:

$$\langle\, \mathbf{r}^{(k)}, \tilde{\mathbf{x}}' \,\rangle = \int_{t_{-k}}^{t_{1-k}} (\mathbf{r}^{(k)})^H \tilde{\mathbf{x}}'\, dt. \tag{17}$$

Then, the necessary adjoint identity is given by

$$\langle\, \mathbf{r}^{(k)}, \mathcal{L}\tilde{\mathbf{x}}' \,\rangle = \langle\, \mathcal{L}^*\mathbf{r}^{(k)}, \tilde{\mathbf{x}}' \,\rangle + b^{(k)}. \tag{18a}$$

Using the definition of the operator $\mathcal{L}$ given by (15) and the appropriate integration by parts, it is easily shown that

$$\mathcal{L}^*\mathbf{r}^{(k)} = -\frac{d\mathbf{r}^{(k)}(t)}{dt} - A^H \mathbf{r}^{(k)}(t), \tag{18b}$$

$$b^{(k)} = (\mathbf{r}^{(k)}_{1-k})^H \tilde{\mathbf{x}}'_{1-k} - (\mathbf{r}^{(k)}_{-k})^H \tilde{\mathbf{x}}'_{-k}. \tag{18c}$$

Returning to the perturbed cost function, (16) can be rewritten as:

$$\mathcal{J}'(\mathbf{u}') = (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K})^H \mathcal{P}^{-1}_{-K|-K}\, \mathbf{u}' - \mathcal{J}'_1 - \sum_{k=-K}^{-1} (\mathbf{y}_k - H\tilde{\mathbf{x}}_k)^H \mathcal{R}^{-1} H\tilde{\mathbf{x}}'_k, \tag{19a}$$

$$\mathcal{J}'_1 = \big[H^H \mathcal{R}^{-1}(\mathbf{y}_0 - H\tilde{\mathbf{x}}_0)\big]^H \tilde{\mathbf{x}}'_0. \tag{19b}$$

Looking at the adjoint defined over the last interval, $\mathbf{r}^{(1)}(t)$, the following criteria is enforced:

$$\mathcal{L}^*\mathbf{r}^{(1)} = 0 \quad \Rightarrow \quad \langle\, \mathcal{L}^*\mathbf{r}^{(1)}, \tilde{\mathbf{x}}' \,\rangle = 0, \tag{20a}$$

$$\mathbf{r}^{(1)}_0 = H^H \mathcal{R}^{-1}(\mathbf{y}_0 - H\tilde{\mathbf{x}}_0). \tag{20b}$$

Substituting (15) and (20a) into (18a) for $k = 1$ gives:

$$\begin{aligned} b^{(1)} &= 0 \\ &\Rightarrow\ (\mathbf{r}^{(1)}_0)^H \tilde{\mathbf{x}}'_0 - (\mathbf{r}^{(1)}_{-1})^H \tilde{\mathbf{x}}'_{-1} = 0, \\ &\Rightarrow\ \big[H^H \mathcal{R}^{-1}(\mathbf{y}_0 - H\tilde{\mathbf{x}}_0)\big]^H \tilde{\mathbf{x}}'_0 = (\mathbf{r}^{(1)}_{-1})^H \tilde{\mathbf{x}}'_{-1}, \end{aligned} \tag{21}$$

which allows us to re-express $\mathcal{J}'_1$ in (19b) as

$$\mathcal{J}'_1 = (\mathbf{r}^{(1)}_{-1})^H \tilde{\mathbf{x}}'_{-1}. \tag{22}$$

Note that (20a) and (20b) give the full evolution equation and starting condition for the adjoint $\mathbf{r}^{(1)}$ defined on the interval $t \in [t_{-1}, t_0]$. Hence, a backward march over this interval will lead to the term $\mathbf{r}^{(1)}_{-1}$ contained in (22).

The perturbed cost function (19a) can now be rewritten such that

$$\mathcal{J}'(\mathbf{u}') = (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K})^H \mathcal{P}^{-1}_{-K|-K}\, \mathbf{u}' - \mathcal{J}'_2 - \sum_{k=-K}^{-2} (\mathbf{y}_k - H\tilde{\mathbf{x}}_k)^H \mathcal{R}^{-1} H\tilde{\mathbf{x}}'_k, \tag{23a}$$

$$\Rightarrow \quad \mathcal{J}'_2 = \big[H^H \mathcal{R}^{-1}(\mathbf{y}_{-1} - H\tilde{\mathbf{x}}_{-1}) + \mathbf{r}^{(1)}_{-1}\big]^H \tilde{\mathbf{x}}'_{-1}. \tag{23b}$$

Enforcing the following conditions [cf. (20)] for the adjoint on this interval, $\mathbf{r}^{(2)}(t)$,

$$\mathcal{L}^*\mathbf{r}^{(2)} = 0, \tag{24a}$$

$$\mathbf{r}^{(2)}_{-1} = H^H \mathcal{R}^{-1}(\mathbf{y}_{-1} - H\tilde{\mathbf{x}}_{-1}) + \mathbf{r}^{(1)}_{-1}, \tag{24b}$$

it can be shown via a derivation similar to (21) that

$$\mathcal{J}'_2 = (\mathbf{r}^{(2)}_{-2})^H \tilde{\mathbf{x}}'_{-2}, \tag{25}$$

which is of identical form as (22). Thus, it follows that each of the adjoints can be defined in such a way as to collapse the sum in the perturbed cost function (16) as above, until the last adjoint equation $\mathbf{r}^{(K)}$ reduces the perturbed cost function to the following:

$$\mathcal{J}'(\mathbf{u}') = (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K})^H \mathcal{P}^{-1}_{-K|-K}\, \mathbf{u}' - (\mathbf{r}^{(K)}_{-K})^H \tilde{\mathbf{x}}'_{-K} - (\mathbf{y}_{-K} - H\tilde{\mathbf{x}}_{-K})^H \mathcal{R}^{-1} H\tilde{\mathbf{x}}'_{-K}, \tag{26}$$

with the adjoints over the K intervals being defined as:

$$\begin{aligned} &\frac{d\mathbf{r}^{(1)}(t)}{dt} = -A^H \mathbf{r}^{(1)}(t), && \mathbf{r}^{(1)}_0 = 0 && + H^H \mathcal{R}^{-1}(\mathbf{y}_0 - H\tilde{\mathbf{x}}_0), \\ &\frac{d\mathbf{r}^{(2)}(t)}{dt} = -A^H \mathbf{r}^{(2)}(t), && \mathbf{r}^{(2)}_{-1} = \mathbf{r}^{(1)}_{-1} && + H^H \mathcal{R}^{-1}(\mathbf{y}_{-1} - H\tilde{\mathbf{x}}_{-1}), \\ &\qquad\vdots && \qquad\vdots \\ &\frac{d\mathbf{r}^{(K)}(t)}{dt} = -A^H \mathbf{r}^{(K)}(t), && \mathbf{r}^{(K)}_{1-K} = \mathbf{r}^{(K-1)}_{1-K} && + H^H \mathcal{R}^{-1}(\mathbf{y}_{1-K} - H\tilde{\mathbf{x}}_{1-K}). \end{aligned} \tag{27}$$

Upon further examination, the system of adjoints (27) all have the same form. Each adjoint variable $\mathbf{r}^{(k+1)}$ is endowed with a starting condition that is the final condition of the adjoint march $\mathbf{r}^{(k)}$ plus a correction due to the discrete measurement $\mathbf{y}_{-k}$ at the measurement time t_{-k}. Thus, the total adjoint march can be thought of as one continuous-time march of a single adjoint variable $\mathbf{r}(t)$ backward over the window $[t_{-K}, t_0]$, with discrete "jumps" in $\mathbf{r}$ at each

measurement time t_k. That is, (27) can be rewritten as:

$$\frac{d\mathbf{r}(t)}{dt} = -A^H \mathbf{r}(t), \tag{28a}$$

which is marched backward over the entire interval $t \in [t_{-K}, t_0]$ with $\mathbf{r}_0 = 0$. At the measurement times (t_k for $k \in M$) the adjoint is updated such that

$$\mathbf{r}_k \leftarrow \mathbf{r}_k + H^H \mathcal{R}^{-1}(\mathbf{y}_k - H\tilde{\mathbf{x}}_k). \tag{28b}$$

Note that this update is performed right at the beginning of the march, at t_0, and also right at the end of the march, at t_{-K}, as well at all the measurement times in between. Then, this definition of the adjoint can be substituted into (26) to give:

$$\mathcal{J}'(\mathbf{u}') = (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K})^H \mathcal{P}^{-1}_{-K|-K} \mathbf{u}' - \mathbf{r}^H_{-K} \tilde{\mathbf{x}}'_{-K}, \tag{29}$$

$$\Rightarrow \quad \mathcal{J}'(\mathbf{u}') = \left[\mathcal{P}^{-1}_{-K|-K}(\mathbf{u} - \bar{\mathbf{x}}_{-K|-K}) - \mathbf{r}_{-K} \right]^H \mathbf{u}', \tag{30}$$

where (30) is found by noting that $\tilde{\mathbf{x}}'_{-K} = \mathbf{u}'$. Then finally, from (13) and (30), the gradient sought may be written as:

$$\bigtriangledown \mathcal{J}(\mathbf{u}) = \mathcal{P}^{-1}_{-K|-K}(\mathbf{u} - \bar{\mathbf{x}}_{-K|-K}) - \mathbf{r}_{-K}. \tag{31}$$

The resulting gradient[5] can then be used iteratively to update the current estimate via a suitable minimization algorithm (steepest descent, conjugate gradient, limited-memory BFGS, etc.).

Being vector based [see (28), (31)] makes 4DVar well suited for multiscale problems, and as a result is currently used extensively by the weather forecasting community. However, it has several key disadvantages. Most significantly, upon convergence, the algorithm provides an updated mean estimate $\bar{\mathbf{x}}_{-K|0}$, but provides no clear formula for computing the updated estimate uncertainty covariance or its inverse, $\mathcal{P}^{-1}_{-K|0}$. That is, the statistical distribution of the estimate probability is not contained in the output of a traditional 4DVar algorithm. It can be shown that, upon full convergence for a linear system, the resulting analysis covariance $\mathcal{P}_{-K|0}$ is simply the Hessian of the original cost function (11) [see, e.g., Bouttier & Courtier (2002)]. However, this is merely an analytical curiosity; computing the analysis covariance in this fashion requires as much matrix algebra as would be required to propagate a sequential filter through the entire variational window, defeating the purpose of the vector-based method.

Additionally, as posed above, the width of the variational window is fixed in the traditional 4DVar formulation. Thus, the cost function and associated *n*-dimensional minimization surface are also constant throughout the iterations. For nonlinear systems, especially chaotic systems, this makes traditional 4DVar extremely sensitive to initial conditions. Because of the nature of these systems, the optimization surfaces are highly irregular and fraught with local minima. The gradient-based algorithms associated with 4DVar are only guaranteed to converge to local minima. Thus, if the initial background estimate is located in the region of attraction of one of these local minima, the solution of the 4DVar algorithm will tend to converge to a suboptimal estimate.

Lastly, due to the complex nature of multiscale fluid systems, the computation time required for full convergence of the fixed-horizon 4DVar algorithm is usually non-negligible when compared with the characteristic time scales of the system, even though many of the largest purpose-built supercomputers ever built have been fully dedicated to weather-forecasting problems. As iterations of 4DVar over a fixed horizon proceed, one is effectively solving more and more accurately a problem which, as time bears on, one cares less and less about. When the 4DVar algorithm finally converges, the estimate so determined is for a time that has already slipped far into the past, and is of reduced relevance for producing an accurate forecast.

[5] Omitted in this gradient derivation is the substantial flexibility in the choice of the gradient definition (13) and the duality pairing (17). There is freedom in the choice of these inner products (e.g. by incorporating derivative and/or integral operators as well as weighting factors) that can serve to better precondition the optimization problem at hand without affecting its minimum points. This ability to precondition the adjoint problem is discussed at length in Protas et al. (2004).

3 The EnVE Algorithm

The new Ensemble Variational Estimation (EnVE) algorithm is now presented as a consistent hybrid of the two aforementioned assimilation schemes, EnKF and 4DVar. A detailed description of the theoretical aspects of EnVE is first given in Section 3. The practical implementation details of EnVE are then highlighted in Section 4. As explored further in Sections 5 and 6, EnVE is a consistent, receding-horizon, multiscale-in-time assimilation technique which revisits past measurements in light of new data and keeps track of the estimate uncertainty at each step of the algorithm.

Assume, without loss of generality, that an EnKF estimate $\widehat{X}_{-j|-j}$ exists[6] at some past time t_{-j}. This ensemble represents an estimate at time t_{-j} based on measurements up to and including $\mathbf{y}_{-j}$. At this point, available measurements up to t_0 are considered. The EnVE algorithm is initialized via a traditional sequential march of the EnKF up to the time of the most recent measurement, t_0 (see Figure 2). This provides an ensemble estimate at the present time, $\widehat{X}_{0|0}$, and all of its corresponding implied statistics. The mean of this estimate is denoted $\bar{\mathbf{x}}_{0|0}$, and is found by taking the average of all the ensemble members. This estimate at time t_0 is based, in a Kalman-like manner, on all measurements up to and including the present time. Doing a traditional Kalman-like march of this sort would, for an adequate number of ensemble members and a linear system, produce the optimal estimate at t_0. However, errors due to the nonlinearity of the chaotic system and approximations due to the finite size of the ensemble ultimately lead to a suboptimal estimate via the EnKF approach.

For forecasting applications, the most important estimate is the one at the most recent measurement time, t_0, because it is this which is used as an initial condition for any forecasting calculation. With a linear system, any type of smoothing at this stage in the EnKF algorithm would have no effect on the estimate at t_0. The smoother would simply reduce the error in the past estimates, for times $t < t_0$, using the information in the observations between t and t_0. However, for a nonlinear system, smoothing affects the entire estimate trajectory, even the most recent estimate at t_0. This is due to the dependence of the evolution of the estimate uncertainty on the trajectory of the estimate itself. For a linear system, the covariance propagation is independent of trajectory, but for a nonlinear system, changes in a past estimate (via smoothing) will impact the future trajectory of the estimate and its associated covariance. This motivates the consistent revisiting of past measurements in light of new data in order to improve the resulting forecast.

[6] Upon, startup, a large initial spread of the ensembles should be used to indicate substantial uncertainty of the initial condition. This can be accomplished by running the EnKF for a period of time open loop (that is, without any feedback from the measurements).

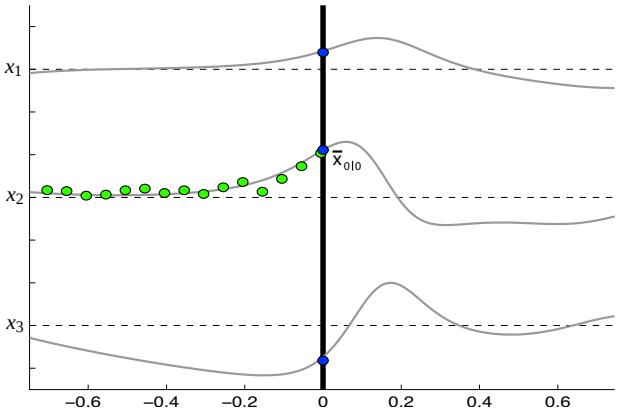

Figure 2. EnVE is initialized by marching a traditional EnKF forward through the available observations, making the appropriate updates. This provides an up-to-date estimate of the current state of the system, $\bar{\mathbf{x}}_{0|0}$, based upon all available measurements. At this point, it may be beneficial to revisit past measurements to update the trajectory of the estimate in light of the more recent measurements. For visualization purposes, EnVE is applied here to the Lorenz equation with noisy measurements of one of its three components.

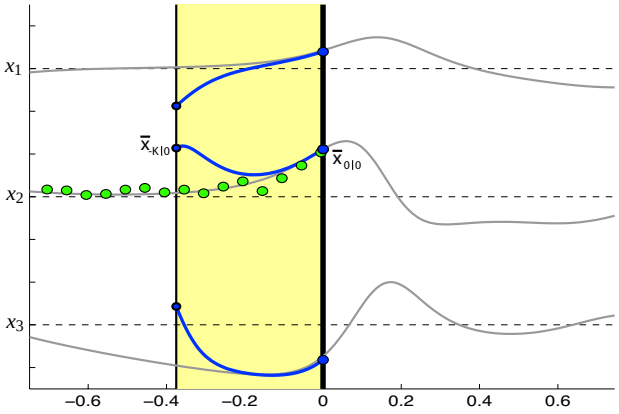

Figure 3. To determine the accuracy of the current estimate (that is, its correlation with the recent measurements), the ensemble at the present time is marched backward using the system equations until the trajectory of the ensemble mean is deemed significantly divergent from the observations. This gives the current best estimate $\bar{x}_{-K|0}$ at the past time t_{-K}.

To this end, the ensemble $\widehat{X}_{0|0}$ is marched backward, using only the model equations. In so doing, the estimate retains the information captured by the measurements during the forward EnKF march. Thus, any point on this resulting trajectory is conditioned on all available measurements. At the conclusion of this backward march, the ensemble mean and implicit statistics are known at some past time, say t_{-K}.

This retrograde ensemble march is monitored in such a way as to define the width of the observation window for the subsequent variational step of the EnVE algorithm. If the initial estimate at t_0 is poor, then a lot of useful information may be deduced from a small time window containing only a few observations. Including more

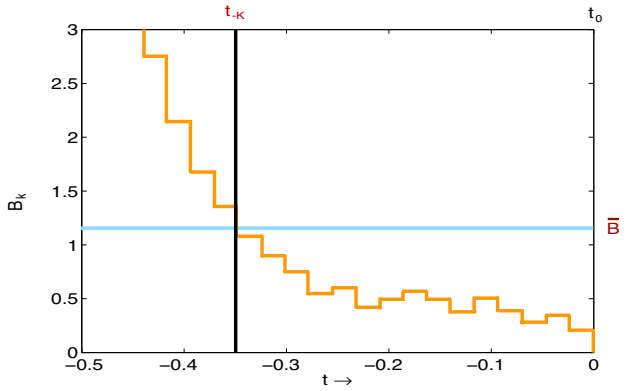

Figure 4. The accumulation of "bias" between the estimate trajectory and the observations is shown as the original estimate is marched backward. Upon reaching a critical bias $\bar{B}$, the retrograde march is stopped. This time t_{-K} defines the width of the subsequent variational window.

observations in this case is superfluous, and in fact unnecessarily increases the complexity of the optimization surface. Conversely, if the initial estimate at t_0 is very accurate, then a significantly longer variational window can, and should, be included in the analysis.

The retrograde ensemble march is thus used to define the window width used in the subsequent variational step by looking at the correlation between the trajectory of the ensemble mean and the recent measurement history (see Figure 3). Poor estimates diverge quickly from the measurements, and should be analyzed with short optimization windows; conversely, accurate estimates march much further back in time before they begin to diverge from the measurements, and should be analyzed with longer optimization windows. To quantify this divergence, a "bias" measure is calculated during the backward march. Mathematically, this bias measure B_k may be defined as

$$B_k = \left\| \sum_{j=0}^{-k} (\mathbf{y}_j - H\bar{\mathbf{x}}_{j|0}) \right\|_1 \quad \text{where} \quad \|\mathbf{z}\| = |z_1| + \cdots + |z_n|, \tag{32}$$

and where the sum is computed by marching $\widehat{X}_{0|0}$ backward from the present time, t_0. Note that this bias measure does not square its argument. As long as the misfit between each measurement and the corresponding quantity in the model is as often positive as it is negative, the net contribution to B_k is nearly zero, and the march continues. Once this misfit is consistently one sign or the other, the bias measure rather suddenly begins to grow (see Figure 4), and the march is terminated. Through experimentation, a critical bias $\bar{B}$ is defined such that the trajectory of the ensemble mean is deemed significantly divergent from the observations past this period. This point defines the left edge of the variational window, t_{-K}, as follows:

$$K = \min\{\, k \mid B_k \geqslant \bar{B} \,\}. \tag{33}$$

With the variational window $[t_{-K}, t_0]$ so defined, the initial best smoothed estimate of the state $\bar{\mathbf{x}}_{-K|0}$ is given as the mean of the ensemble $\widehat{X}_{-K|0}$. At this point, variational methods are used to improve this estimate in a consistent manner. To this end, the traditional 4DVar cost function is defined with a background estimate and covariance at t_{-K}. The background term of the cost function must now be defined carefully, as the correct background term is essential for EnVE to be consistent. In other words, properly defining

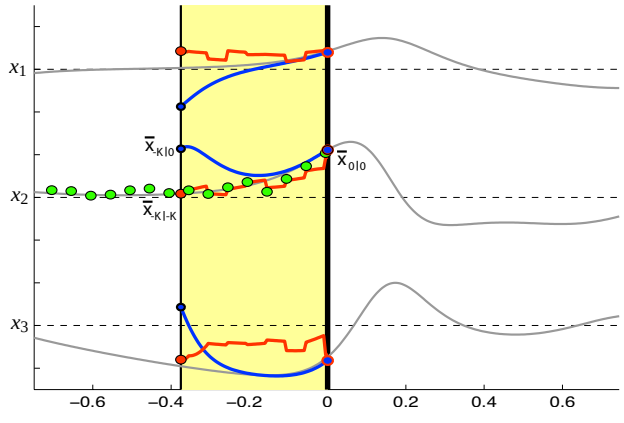

Figure 5. In order to define fully the variational cost function, the background terms at t_{-K} must be recalled. This is done by marching the original ensemble $\widehat{X}_{0|0}$ backward through the window, sequentially removing the effect of each measurement. This march results in a background ensemble $\widehat{X}_{-K|-K}$ at t_{-K}. From this ensemble, the background mean and covariance can be inferred.

the background term in the variational cost function guarantees that erroneous updates are not made by using an observation more than once, and ensures that the result obtained reduces to that obtained by the Kalman Filter in the special case that the system considered happens to be linear.

The correct background term is determined by returning to the original ensemble, $\widehat{X}_{0|0}$, and marching it backward again to t_{-K}, this time *removing* the effects of the measurement updates along the way (see Figure 5). As the EnKF is an approximation of the KF, in order to derive this backward-in-time EnKF, the backward-in-time KF first needs to be understood. To this end, the backward-marching KF equations are now derived that remove the measurement updates in a manner similar to the traditional forward-marching KF equations, which add the measurement updates. Because the KF is considered here, non-singularity of the covariance $\mathcal{P}$ is assumed in this derivation.

To begin, recall the standard KF update for the forward march (both for the mean estimate and the covariance):

$$\bar{\mathbf{x}}_{k|k} = \bar{\mathbf{x}}_{k|k-1} + \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} (\mathbf{y}_k - H \bar{\mathbf{x}}_{k|k-1}), \quad (34)$$

$$\mathcal{P}_{k|k} = \mathcal{P}_{k|k-1} - \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} H \mathcal{P}_{k|k-1}. \quad (35)$$

It is expected that the equation for the backward march will be of similar form. Rearranging terms, and assuming that the measurement $\mathbf{y}_k \in \text{range}(H)$ (i.e. $\mathbf{y}_k = H\mathbf{q}_k$ for some $\mathbf{q}_k \in \Re^n$), gives the following expression:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k} =& \left[I - \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} H\right] \bar{\mathbf{x}}_{k|k-1} \\ &+ \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} H \mathbf{q}_k. \end{aligned} \quad (36)$$

Note that the assumption $\mathbf{y}_k = H\mathbf{q}_k$ for some $\mathbf{q}_k$ is, in practice, not a restrictive assumption, as it only requires that H have linearly independent rows. In most physical systems of interest the measurements are independent, and thus this assumption is valid. Towards the goal of writing the estimate update in terms of the current covariance $\mathcal{P}_{k|k}$ [as opposed to the form of (34), where the update is written in terms of the prediction covariance $\mathcal{P}_{k|k-1}$], the identity $\mathcal{P}_{k|k-1}\left(\mathcal{P}_{k|k-1}\right)^{-1} = I$ is inserted into (36). Rearranging terms gives:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k} =& \left[\mathcal{P}_{k|k-1} - \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} H \mathcal{P}_{k|k-1}\right] \left(\mathcal{P}_{k|k-1}\right)^{-1} \bar{\mathbf{x}}_{k|k-1} \\ &+ \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} H \mathcal{P}_{k|k-1} \left(\mathcal{P}_{k|k-1}\right)^{-1} \mathbf{q}_k. \end{aligned} \quad (37)$$

Substituting for the updated covariance from (35) produces:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k} =& \mathcal{P}_{k|k} \left(\mathcal{P}_{k|k-1}\right)^{-1} \bar{\mathbf{x}}_{k|k-1} \\ &+ \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} H \mathcal{P}_{k|k-1} \left(\mathcal{P}_{k|k-1}\right)^{-1} \mathbf{q}_k. \end{aligned} \quad (38)$$

Adding and subtracting $\mathcal{P}_{k|k-1}\left(\mathcal{P}_{k|k-1}\right)^{-1}\mathbf{q}_k$ to the end of (38) and rearranging gives a similar result for the second term, allowing for a substitution for the updated covariance:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k} =& \mathcal{P}_{k|k} \left(\mathcal{P}_{k|k-1}\right)^{-1} \bar{\mathbf{x}}_{k|k-1} + \overbrace{\mathcal{P}_{k|k-1} \left(\mathcal{P}_{k|k-1}\right)^{-1} \mathbf{q}_k - \mathcal{P}_{k|k-1} \left(\mathcal{P}_{k|k-1}\right)^{-1} \mathbf{q}_k}^{=0} \\ &+ \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} H \mathcal{P}_{k|k-1} \left(\mathcal{P}_{k|k-1}\right)^{-1} \mathbf{q}_k \\ =& \mathcal{P}_{k|k} \left(\mathcal{P}_{k|k-1}\right)^{-1} \bar{\mathbf{x}}_{k|k-1} + \mathbf{q}_k \\ &- \left[\mathcal{P}_{k|k-1} - \mathcal{P}_{k|k-1} H^H (H \mathcal{P}_{k|k-1} H^H + \mathcal{R})^{-1} H \mathcal{P}_{k|k-1}\right] \left(\mathcal{P}_{k|k-1}\right)^{-1} \mathbf{q}_k \\ =& \mathcal{P}_{k|k} \left(\mathcal{P}_{k|k-1}\right)^{-1} \bar{\mathbf{x}}_{k|k-1} + \left[I - \mathcal{P}_{k|k} \left(\mathcal{P}_{k|k-1}\right)^{-1}\right] \mathbf{q}_k. \end{aligned} \quad (39)$$

Returning to (35), the matrix inversion lemma can be used to solve for the backward covariance update:

$$\left(\mathcal{P}_{k|k}\right)^{-1} = \left(\mathcal{P}_{k|k-1}\right)^{-1} + H^H \mathcal{R}^{-1} H, \quad (40)$$

$$\left(\mathcal{P}_{k|k-1}\right)^{-1} = \left(\mathcal{P}_{k|k}\right)^{-1} - H^H \mathcal{R}^{-1} H, \quad (41)$$

$$\mathcal{P}_{k|k-1} = \mathcal{P}_{k|k} - \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} H \mathcal{P}_{k|k}. \quad (42)$$

Note the similarity between adding the measurement update in (35) and removing the measurement update in (42). For the estimate update, the following identity is determined via (41)

$$\mathcal{P}_{k|k} \left(\mathcal{P}_{k|k-1}\right)^{-1} = I - \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} H. \quad (43)$$

Looking again at the estimate update (39), the identity (43) can be substituted to simplify the right-hand side. The assumption $\mathbf{y}_k = H\mathbf{q}_k$ is then reinserted to produce a closed-form expression for the update in terms of the updated covariance only:

$$\bar{\mathbf{x}}_{k|k} = \left[I - \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} H\right] \bar{\mathbf{x}}_{k|k-1} + \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} H \mathbf{q}_k, \quad (44)$$

$$\bar{\mathbf{x}}_{k|k} = \left[I - \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} H\right] \bar{\mathbf{x}}_{k|k-1} + \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} \mathbf{y}_k, \quad (45)$$

$$\bar{\mathbf{x}}_{k|k} = \bar{\mathbf{x}}_{k|k-1} + \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} (\mathbf{y}_k - H \bar{\mathbf{x}}_{k|k-1}). \quad (46)$$

The form given in (46) is useful because the updated covariance is all that is available when the update is reversed. Additionally, note the striking similarity of the update gain in (46) to the classical continuous time Kalman filter update equation[7]. Now, (45) can be solved directly for the estimate without the update.

$$\bar{\mathbf{x}}_{k|k-1} = \left[I - \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} H\right]^{-1} \left(\bar{\mathbf{x}}_{k|k} - \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} \mathbf{y}_k\right). \quad (47)$$

[7] It is worth noting that the measurement update equation in the form given in (46) is equivalent to the standard discrete-time update equation (34) for the KF and EKF. The difference is that (46) is written as a function of the current covariance $\mathcal{P}_{k|k}$, as opposed to the typical update in (34) based on the prediction covariance $\mathcal{P}_{k|k-1}$. For the KF and EKF, it is not necessary to update the estimate before the covariance, so a significant computational savings can be realized by doing these updates opposite of the traditional order: first update the covariance using the standard update equation (35), then update the estimate using (46).

Using the matrix inversion lemma, (47) becomes:

$$\bar{\mathbf{x}}_{k|k-1} = \left[I - \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} H\right](\bar{\mathbf{x}}_{k|k} - \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} \mathbf{y}_k). \tag{48}$$

Expanding the product gives:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k-1} =& \bar{\mathbf{x}}_{k|k} - \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} H \bar{\mathbf{x}}_{k|k} - \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} \mathbf{y}_k \\ &+ \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} H \mathcal{P}_{k|k} H^H \mathcal{R}^{-1} \mathbf{y}_k. \end{aligned} \tag{49}$$

The final two terms can be factored, simplified, and rearranged:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k-1} &= \bar{\mathbf{x}}_{k|k} - \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} H \bar{\mathbf{x}}_{k|k} \\ &\quad + \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} \times \\ &\quad \left[-(H \mathcal{P}_{k|k} H^H - \mathcal{R}) + H \mathcal{P}_{k|k} H^H\right] \mathcal{R}^{-1} \mathbf{y}_k \\ &= \bar{\mathbf{x}}_{k|k} - \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} H \bar{\mathbf{x}}_{k|k} \\ &\quad + \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} \mathbf{y}_k \\ &= \bar{\mathbf{x}}_{k|k} + \mathcal{P}_{k|k} H^H (H \mathcal{P}_{k|k} H^H - \mathcal{R})^{-1} (\mathbf{y}_k - H \bar{\mathbf{x}}_{k|k}). \end{aligned} \tag{50}$$

Note the striking similarity between the measurement "downdate" equation (50) and the measurement update equation (34).

Note that (50) is the closed-form analytical expression for removing the effect of a measurement update using the KF. This "downdate" equation, coupled with (42) and the backward marching state equations, can be used while marching the KF backward in time, exactly removing the measurement updates along the way. As (8) is the ensemble representation of (34) for the KF, a similar "downdating" EnKF can be found from the "downdating" KF equation (50):

$$\hat{\mathbf{x}}^j_{k|k-1} = \hat{\mathbf{x}}^j_{k|k} + \mathcal{P}^e_{k|k} H^H (H \mathcal{P}^e_{k|k} H^H - \mathcal{R})^{-1} (\mathbf{d}^j_k - H \hat{\mathbf{x}}^j_{k|k}). \tag{51}$$

This equation governs the "downdates" necessary to reverse the forward march of the ensemble $\widehat{X}_{0|0}$ (determined using updates from all measurements) in order to determine the background ensemble $\widehat{X}_{-K|-K}$ representing, in the linear setting, the background estimate and statistics at t_{-K} containing no information about the observations within the variational window. From this background ensemble, the background mean $\bar{\mathbf{x}}_{-K|-K}$ and background covariance $\mathcal{P}^e_{-K|-K}$ can be extracted.

In the ensemble implementation of the variational step there is an additional somewhat subtle wrinkle to the 4DVar derivation presented in Section 2.3. Recall that the traditional 4DVar cost function (11) measures the misfit between the measurements and the model trajectory $\tilde{\mathbf{x}}(t)$ with $\tilde{\mathbf{x}}_{-K} = \mathbf{u}$. In contrast, during the variational iteration associated with the EnVE algorithm, this mean trajectory is defined as the average of the ensemble trajectories over the window, and therefore is not itself necessarily even a trajectory of the underlying model. That is, with EnVE,

$$\frac{d\hat{\mathbf{x}}^j(t)}{dt} = f(\hat{\mathbf{x}}^j(t), \mathbf{w}^j(t)), \quad \tilde{\mathbf{x}}(t) = \frac{1}{N} \sum_{j=1}^{N} \hat{\mathbf{x}}^j(t). \tag{52}$$

The corresponding cost function $\mathcal{J}(\mathbf{u})$ that is minimized (with respect to $\mathbf{u}$) by EnVE is defined in a similar manner as in traditional 4DVar:

$$\begin{aligned}\mathcal{J}(\mathbf{u}) =& \frac{1}{2} (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K})^H \left(\mathcal{P}^e_{-K|-K}\right)^+ (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K}) \\ &+ \frac{1}{2} \sum_{k=-K}^{0} (\mathbf{y}_k - H\tilde{\mathbf{x}}_k)^H \mathcal{R}^{-1} (\mathbf{y}_k - H\tilde{\mathbf{x}}_k), \end{aligned} \tag{53}$$

where the "optimization variable" $\mathbf{u}$ is the value of the refined state estimate $\tilde{\mathbf{x}}$, given as the average of the ensembles at t_{-K}; that is,

$$\tilde{\mathbf{x}}_{-K} = \frac{1}{N} \sum_{j=1}^{N} \hat{\mathbf{x}}^j_{-K|0} = \mathbf{u}. \tag{54}$$

Note that (53) is consistent with the 4DVar cost function (11). In (53), however, the estimate covariance matrix is replaced by the ensemble estimate covariance matrix. For the multiscale systems of interest, this background covariance is singular. Thus, the pseudoinverse must be used instead. Because the background term of the cost function is consistently defined (in that, in the linear setting, it incorporates no information from the observations within the variational window), the corresponding n-dimensional optimization surface is, in the linear case, identical to what would have been used had no sequential march through those observations been completed.

With the cost function defined appropriately in this manner, a variational iteration can now be performed, similar to 4DVar. With traditional 4DVar, the first iteration is typically initialized using the background term, $\mathbf{u} = \bar{\mathbf{x}}_{-K|-K}$. However, with EnVE, a better estimate than this is already known, namely the smoothed ensemble mean, $\mathbf{u} = \bar{\mathbf{x}}_{-K|0}$. This is one of the strengths of EnVE: it initializes the variational iteration with an estimate that is known to be significantly better than the background. In either case, the optimization surface is identical, but with EnVE, the initial ensemble estimate for $\mathbf{u}$ is much closer to the global minimum than the original background term. Consequently, if any significant improvement can be made upon this initial estimate, it will be discovered in the first variational iteration(s). Further, the initial estimate is more likely to be in the region of attraction of the global minimum, so the probability of erroneous convergence to spurious local minima can be substantially reduced.

In minimizing the cost function, the goal is to shift the first moment statistics of the ensemble without altering the higher moments. To this end, a simple translation of the ensemble as a whole is desired. Consequently, the sensitivity of the cost function (53) with respect to an ensemble translation is examined. As many of the adjoint derivation steps are similar to those described in Section 2.3, only modifications related to the new formulation will be discussed here.

As the mean trajectory can not be perturbed directly, the cost function $\mathcal{J}$, the optimization variable $\mathbf{u}$, and the ensemble trajectories $\hat{\mathbf{x}}^j(t)$ are perturbed to give the perturbed cost function $\mathcal{J}'(\mathbf{u}')$ as:

$$\begin{aligned}\mathcal{J}'(\mathbf{u}') =& (\mathbf{u} - \bar{\mathbf{x}}_{-K|-K})^H (\mathcal{P}^e_{-K|-K})^+ \mathbf{u}' \\ &- \frac{1}{N} \sum_{j=1}^{N} \left[\sum_{k=-K}^{0} (\mathbf{y}_k - H\tilde{\mathbf{x}}_k)^H \mathcal{R}^{-1} H \hat{\mathbf{x}}^{j\prime}_k \right]. \end{aligned} \tag{55}$$

Importantly, the ensemble perturbations $\hat{\mathbf{x}}^{j\prime}_k$ are related to $\mathbf{u}'$ due to the assumption that, at t_{-K}, only a translation of the ensemble will be allowed, i.e.,

$$\hat{\mathbf{x}}^j_{-K}{}' = \mathbf{u}' \qquad \forall j \in [1, N]. \tag{56}$$

The components of the outer summation in (55) over the ensemble members can now be related by defining an individual adjoint variable $\mathbf{r}^j(t)$ for each ensemble member. Similar to 4DVar, the inner summation over the measurement times can be re-expressed–leveraging each adjoint $\mathbf{r}^j(t)$–in a manner identical to Section 2.3, in which a sequence of adjoints are defined over the measurement intervals, and it is seen that the intervals can be compressed into one continuous-time adjoint equation with discrete forcing at the

measurement times. In this manner, with the EnVE implementation, an "ensemble" of N adjoints is defined over the window, with each individual adjoint equation linearized about the trajectory of its corresponding ensemble member as follows:

$$\frac{d\mathbf{r}^j(t)}{dt} = -A\big(\hat{\mathbf{x}}^j(t)\big)^H \mathbf{r}^j(t), \qquad \mathbf{r}^j_0 = 0. \tag{57}$$

At the measurement times, an identical discrete update is made to each adjoint corresponding to the deviation of the ensemble mean from the measurement; i.e., at the measurement times,

$$\mathbf{r}^j_k \leftarrow \mathbf{r}^j_k + H^H \mathcal{R}^{-1}(\mathbf{y}_k - H\tilde{\mathbf{x}}_k), \quad \text{where} \quad \tilde{\mathbf{x}}_k = \frac{1}{N}\sum_{j=1}^{N} \hat{\mathbf{x}}^j_k. \tag{58}$$

Thus, a forward march of the ensemble estimate through the variational window provides the trajectories that will be used to drive the N adjoints backward through the window. At each measurement time, the ensemble of adjoints are all translated by calculating the misfit between the ensemble mean and the corresponding measurement. These parallel marches serve to re-express the inner summation over the measurements in the perturbed cost function (55). Finally using the perturbation equation (56) at t_{-K}, the gradient of the original cost function can be expressed [cf. (31)] as:

$$\bigtriangledown \mathcal{J}(\mathbf{u}) = (\mathcal{P}^e_{-K|-K})^+(\mathbf{u} - \bar{\mathbf{x}}_{-K|-K}) - \frac{1}{N}\sum_{j=1}^{N} \mathbf{r}_{-K}. \tag{59}$$

In other words, the component of the gradient due to the misfit of the ensemble with the measurements is simply the average of the contributions from each individual adjoint at t_{-K}. Note that, in the linear setting, computing the gradient using multiple adjoints in this manner is equivalent to forcing a single adjoint about the mean trajectory, as–in this special case only–the trajectory of the ensemble mean is the same as the mean of the ensemble trajectories.

With 4DVar, as described previously, the estimate $\mathbf{u} = \tilde{\mathbf{x}}_{-K}$ would be marched forward using the model over the variational window. This trajectory needs to be stored or checkpointed, because it drives the subsequent backward march of the adjoint over the same window. For large systems, this presents a significant computational challenge. With EnVE, however, this trajectory is determined via a backward march of the ensemble (see Figure 3). Since the background term and the width of the variational window do not need to be known before the adjoint march begins, this facilitates a simultaneous march of all three systems (the ensemble estimate without the measurement "downdates", the ensemble of adjoints, and the ensemble estimate with measurement "downdates") from t_0 until the mean of the estimate diverges sufficiently from the observations (at t_{-K}), as defined by the bias measure B_k [see (32)]. The computational benefits of such parallel marches are more fully examined in Section 6.3. Because they are marched in parallel, the ensemble member trajectories are immediately available to drive the adjoint computations "on the fly", and the storage challenge normally associated with adjoint-based methods is eliminated. At the conclusion of the backward march, the window width, the appropriate background term, and the adjoint at t_{-K} are determined, and thus the gradient (59) of the variational cost function may be extracted.

Note that the evaluation of this gradient requires the computation of the pseudoinverse of the ensemble background covariance. Fortunately, exploiting the structure of the ensemble framework, this pseudoinverse can be computed efficiently even for high-dimensional systems (the specifics of this gradient calculation are discussed in Section 4). This gradient, along with a suitable line minimization algorithm, is then used to update each ensemble member (and consequently, the ensemble mean):

$$\hat{\mathbf{x}}^j_{-K|0} \leftarrow \hat{\mathbf{x}}^j_{-K|0} - \alpha \bigtriangledown \mathcal{J}(\mathbf{u}). \tag{60}$$

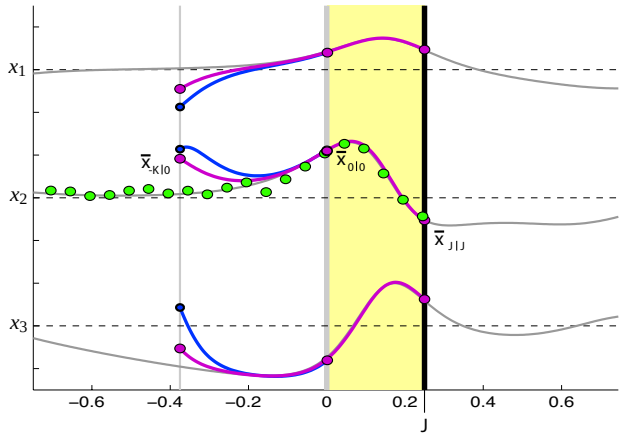

Figure 6. Upon completion of a variational step, the improved ensemble estimate $\widehat{X}_{-K|0}$ at t_{-K} is propagated forward to the old present time t_0. No measurement updates are done during this march, as the observations have already been accounted for. Upon reaching t_0, the new ensemble estimate $\widehat{X}_{0|0}$ is marched sequentially forward using the EnKF to account for any additional measurements received during the computation time required for the previous variational step, and the algorithm is repeated.

Recall from (56) that the derivation above assumed that the final estimate ensemble was obtained simply by shifting the initial (smoothed) estimate ensemble $\widehat{X}_{-K|0}$. In fact, with an adjusted estimate of this sort, a modified if not improved covariance $\mathcal{P}^e_{-K|0}$ would be expected as well. However, as variational methods do not appear to provide a means for tracking these changes, the EnVE algorithm proposed here simply uses this shifted ensemble representation, which is a bit conservative. Note, though, that this is a significant improvement over 4DVar, in which rigorous methods to march $\mathcal{P}$ are essentially unavailable. In contrast, with EnVE, the covariance associated with the original smoothed estimate is available, so it can be utilized. Though this is a conservative estimate of the covariance that does not account for the correction to the estimate due to the variational step, it correctly captures the main features of the covariance matrix, including the principle directions of estimate uncertainty.

To cycle the algorithm, the updated ensemble is marched forward to t_0 (see Figure 6). Note that the ensemble already accounts for the measurements in this window, so each ensemble member is propagated forward using the system equations only, with no additional measurement updates. This gives an improved estimate at t_0, $\widehat{X}_{0|0}$. During the time taken to complete this variational step, some new measurements $\{\mathbf{y}_1 \cdots \mathbf{y}_J\}$ will usually become available. The ensemble $\widehat{X}_{0|0}$ can thus be marched forward further now, using the EnKF to account for these new measurements, until the new present time t_J is reached. At this point, time is reset, $t_0 \leftarrow t_J$, and the algorithm is repeated.

Note that a significant computational burden can be avoided by storing the updated estimate at the previous present time, $\widehat{X}_{0|0}$. This point can serve as a more convenient starting point for determining the subsequent background term of the variational cost function, as opposed to using $\widehat{X}_{J|J}$. Depending on the relative

widths of the subsequent variational window and the time elapsed during the current variational step, starting from $\widehat{X}_{0|0}$ instead of $\widehat{X}_{J|J}$ will result in either a shorter backward EnKF march or possibly even a forward EnKF march to the left edge of the new variational window. This simple storage trick reduces the computational cost of the algorithm significantly and shortens (or removes altogether) one of the backward-in-time marches of the estimate ensemble. Note that these backward-in-time marches are ill posed if the ODE system is derived from a PDE with a diffusive component. However, using appropriate regularization [see, e.g., Lattès & Lions (1969), Protas et al. (2004)], such backward-in-time marches can be reasonably well approximated over short time horizons at their larger length scales. Curiously, as a consequence of these backward-in-time marches of the estimate called for by the algorithm, EnVE appears to be most naturally suited for *high*-Reynolds number systems (without a dominant diffusive component at the length scales of interest).

For relatively small ODE systems of dimension n with a relatively large number of ensemble members, $N > n$, $\mathcal{P}^e$ is invertible, and the EnKF "downdate" (51) is well defined. For such small systems, the subsequent variational windows can in fact overlap, as called for by the algorithm described above.

For high-dimensional discretizations of multiscale PDE systems, on the other hand, only a relatively few number of ensemble members are numerically tractable (i.e., $N \ll n$). For such systems, the ensemble covariance matrix is rank deficient, and its singularity leads to a breakdown in the derivation of (51). As a result, no fully consistent backward EnKF march with measurement "downdates" appears possible. By saving the estimate at the previous present time, $\widehat{X}_{0|0}$, a lower limit is thus set on the left edge of the subsequent variational window, and the background term may instead be determined via a *forward* march from $\widehat{X}_{0|0}$. Hence, for the multiscale systems of interest, it appears to be necessary that the variational windows, from one iteration to the next, do not overlap.

The EnVE algorithm is now summarized:

(i) Given the estimate ensemble at some past time, each ensemble member is marched forward to the present time t_0 with sequential updates at each measurement consistent with the EnKF framework. At this point it is beneficial to revisit old measurements to refine further the current estimate $\overline{\mathbf{x}}_{0|0}$, the ensemble mean.

(ii) The current ensemble $\widehat{X}_{0|0}$ is marched backward until the mean trajectory (the average of the ensemble trajectory) diverges significantly from the measurements. This march determines the number of measurements K in the variational window to be used; poor estimates will have small windows, whereas accurate estimates will have larger windows that incorporate a longer measurement history. Concurrently, the appropriate adjoint ensemble is also marched backward, with discrete forcing updates based on the misfit between the estimates and the corresponding observations. In order to refine the ensemble-mean estimate of the system, a variational iteration is now initialized to optimize this estimate at t_{-K}.

(iii) The current ensemble $\widehat{X}_{0|0}$ is marched backward again, this time removing the measurement updates. This march is used to determine the ensemble-averaged value of the "background state" $\overline{\mathbf{x}}_{-K|-K}$, as well as the "background covariance" $\mathcal{P}^e_{-K|-K}$. As in 4DVar, a cost function over the window of interest, $[t_{-K}, t_0]$, is defined with this background term to summarize the information gleaned from measurements prior to t_{-K}. This cost function is then minimized using standard 4DVar-like techniques. Typically, only one iteration step is performed: the gradient is determined using the (previously calculated) adjoint, and a step size is determined using a suitable line minimization algorithm.

(iv) The line minimization serves to shift the smoothed ensemble estimate $\widehat{X}_{-K|0}$ around an improved mean at t_{-K}. This resulting improved ensemble is propagated forward using the system model without measurement updates. Once the old present time t_0 has been reached, new measurements are available, so the algorithm is repeated from (i), marching the EnKF to the new present time t_J. The ensemble estimate $\widehat{X}_{0|0}$ is saved to simplify computation of the background term during the subsequent variational step.

4 Numerical Implementation in an MPI setting

Some of the numerical issues with regards to the implementation of EnVE are now addressed. The numerical methods available for marching both the state and adjoint, though sometimes nontrivial, are fairly standard. The regularization of the retrograde marches of ill-posed problems (derived from diffusive PDEs) is an active area of research [see Lattès & Lions (1969), Protas et al. (2004)], and deserves even closer consideration in future work. Instead of exploring these issues, this section will focus specifically on the parallel implementation of the EnKF update equations using the Message Passing Interface (MPI), allowing for uniform load distribution on, and minimal communication between, the massively parallel computational resources required to apply the EnVE algorithm to multiscale systems.

In general, the ensemble $\widehat{X}_{i|k}$ is comprised of $N \ll n$ ensemble members. Each of these ensemble members $\hat{\mathbf{x}}^j_{i|k}$ is located on its own processor (or processors) with a corresponding process number. In practice, for testing purposes, an additional process is also used for the "truth" model simulation, which is done in parallel with the EnKF march. Thus, the MPI environment is constructed of $N+1$ processes, with process j denoted by p^j. For convenience, the "truth" model is run on p^0, while each ensemble member $\hat{\mathbf{x}}^j_{i|k}$ is run on its corresponding process, p^j.

The EnKF consists of two main steps: a forward march of the ensemble to predict the estimate at the next measurement, and an appropriate update to the forecasted estimate due to each measurement. Recall that the discretized system of interest is given by (1a) and (1b). The forecasting step of the EnKF is the march from $\widehat{X}_{k-1|k-1}$ to $\widehat{X}_{k|k-1}$ (not including the measurement update). In the MPI setting, this is done by simply marching each ensemble member forward in time–using an appropriate time-stepping algorithm–according to the governing equation:

$$\frac{d\hat{\mathbf{x}}^j(t)}{dt} = f(\hat{\mathbf{x}}^j(t), \mathbf{w}(t)). \tag{61}$$

The disturbances $\mathbf{w}(t)$ are modeled appropriately using a reversible random-number generator [see Colburn & Bewley (2008)], and each ensemble member is disturbed independently from the other ensemble members. In an MPI setting, the computation time of each process is assumed independent from the other processes. Hence, the time required to propagate the N ensemble members in this framework is equivalent to a single simulation on a single processor.

Next, the measurement update at time t_k must be performed. To update the ensemble $\widehat{X}_{k|k-1}$ to reflect the newest measurement

(thereby giving $\widehat{X}_{k|k}$), a corresponding update must be done on each individual ensemble member as follows:

$$\hat{\mathbf{x}}^j_{k|k} = \hat{\mathbf{x}}^j_{k|k-1} + \mathcal{P}^e_{k|k-1} H^H (H \mathcal{P}^e_{k|k-1} H^H + \mathcal{R})^{-1} (\mathbf{d}^j_k - H \hat{\mathbf{x}}^j_{k|k-1}). \quad (62)$$

To evaluate this equation, the three main components of the update are first developed independently as:

$$\hat{\mathbf{x}}^j_{k|k} = \hat{\mathbf{x}}^j_{k|k-1} + L^{(1)}_k \left(L^{(2)}_k\right)^{-1} \mathbf{z}^j_k, \quad (63)$$

$$L^{(1)}_k = \mathcal{P}^e_{k|k-1} H^H \qquad L^{(1)}_k \in \Re^{n\times m}, \quad (64)$$

$$L^{(2)}_k = H \mathcal{P}^e_{k|k-1} H^H + \mathcal{R} \qquad L^{(2)}_k \in \Re^{m\times m}, \quad (65)$$

$$\mathbf{z}^j_k = \mathbf{d}^j_k - H \hat{\mathbf{x}}^j_{k|k-1} \qquad \mathbf{z}^j_k \in \Re^m. \quad (66)$$

Note that the matrices $L^{(1)}_k$ and $L^{(2)}_k$ depend upon the entire ensemble.

First, examine the structure of $\mathcal{P}^e_{k|k-1}$. This covariance is built up from the individual ensemble members such that:

$$\begin{aligned}
\mathcal{P}^e_{k|k-1} =& \frac{1}{N-1} \left[(\hat{\mathbf{x}}^1_{k|k-1} - \overline{\mathbf{x}}_{k|k-1}) \ \cdots \ (\hat{\mathbf{x}}^N_{k|k1} - \overline{\mathbf{x}}_{k|k-1}) \right] \times \\
& \left[(\hat{\mathbf{x}}^1_{k|k-1} - \overline{\mathbf{x}}_{k|k-1}) \ \cdots \ (\hat{\mathbf{x}}^N_{k|k-1} - \overline{\mathbf{x}}_{k|k-1}) \right]^H \\
=& \frac{1}{N-1} \left[\delta\hat{\mathbf{x}}^1_{k|k-1} \ \cdots \ \delta\hat{\mathbf{x}}^N_{k|k-1} \right] \left[\delta\hat{\mathbf{x}}^1_{k|k-1} \ \cdots \ \delta\hat{\mathbf{x}}^N_{k|k-1} \right]^H, \\
\Rightarrow \quad \mathcal{P}^e_{k|k-1} =& \frac{1}{N-1} \sum_{j=1}^{N} \delta\hat{\mathbf{x}}^j_{k|k-1} (\delta\hat{\mathbf{x}}^j_{k|k-1})^H. \quad (67)
\end{aligned}$$

Note that $\mathcal{P}^e_{k|k-1} \in \Re^{n\times n}$; for high-dimensional systems, building up this matrix is computationally intractable but, as shown below, unnecessary in the implementation if the terms are computed in the appropriate order. As is seen in (67), the covariance can be computed as a sum of outer products of the deviations of each ensemble member from the ensemble mean (that is, of the ensemble state perturbation vectors $\delta\hat{\mathbf{x}}^j_{k|k-1}$). Thus, (64) can be written:

$$\begin{aligned}
L^{(1)}_k =& \mathcal{P}^e_{k|k-1} H^H \\
=& \left(H \mathcal{P}^e_{k|k-1}\right)^H \\
=& \frac{1}{N-1} \left(H \sum_{j=1}^{N} \delta\hat{\mathbf{x}}^j_{k|k-1} (\delta\hat{\mathbf{x}}^j_{k|k-1})^H\right)^H \\
=& \frac{1}{N-1} \sum_{j=1}^{N} \delta\hat{\mathbf{x}}^j_{k|k-1} (H \delta\hat{\mathbf{x}}^j_{k|k-1})^H, \\
\Rightarrow \quad L^{(1)}_k =& \frac{1}{N-1} \sum_{j=1}^{N} \delta\hat{\mathbf{x}}^j_{k|k-1} (\delta\hat{\mathbf{y}}^j_{k|k-1})^H, \quad (68)
\end{aligned}$$

where $H\delta\hat{\mathbf{x}}^j_{k|k-1} = \delta\hat{\mathbf{y}}^j_{k|k-1} \in \Re^m$ is the ensemble output perturbation vector. The matrix H is the linearization of the output operator $h : \Re^n \to \Re^m$. Note that, for the multiscale chaotic systems of interest, $m \ll n$ (that is, the number of measurements is much smaller than the dimension of the state), so the storage and communication of the output perturbation vectors $\delta\hat{\mathbf{y}}^j_{k|k-1}$ can be assumed to be negligible compared to the storage and communication of the state and state perturbation vectors. At this point, locally on each process p^j, the ensemble state perturbation $\delta\hat{\mathbf{x}}^j_{k|k-1}$ must be computed along with the ensemble output perturbation $\delta\hat{\mathbf{y}}^j_{k|k-1}$.

Similarly, the first term in $L^{(2)}_k$, namely $H \mathcal{P}^e_{k|k-1} H^H$, can be computed in a manner consistent with $L^{(1)}_k$, exploiting the structure of the ensemble covariance matrix.

$$\begin{aligned}
H \mathcal{P}^e_{k|k-1} H^H =& H \left(\frac{1}{N-1} \sum_{j=1}^{N} \delta\hat{\mathbf{x}}^j_{k|k-1} (\delta\hat{\mathbf{x}}^j_{k|k-1})^H \right) H^H \\
=& \frac{1}{N-1} \sum_{j=1}^{N} (H \delta\hat{\mathbf{x}}^j_{k|k-1}) (H \delta\hat{\mathbf{x}}^j_{k|k-1})^H, \\
\Rightarrow \quad H \mathcal{P}^e_{k|k-1} H^H =& \frac{1}{N-1} \sum_{j=1}^{N} \delta\hat{\mathbf{y}}^j_{k|k-1} (\delta\hat{\mathbf{y}}^j_{k|k-1})^H. \quad (69)
\end{aligned}$$

This term is calculated as a sum over all the processes of the outer product of the ensemble output perturbation with itself (recall that this vector has already been computed on each process). In addition to the $H \mathcal{P}^e_{k|k-1} H^H$ term, $L^{(2)}_k$ contains the measurement covariance matrix $\mathcal{R}$. This matrix, in general, may be a function of time, but a model for $\mathcal{R}$ is assumed to be known.

The structure of many MPI clusters facilitates reasonably efficient all-to-all communication (in which data is passed from every node to every other node in the cluster at the same time). For instance, in a cluster with a toroidal switchless interconnect, all-to-all communication is only slightly more expensive than one-to-all communication (in which one node sends data to ever other node). This is because, in a switchless interconnect torus, during one-to-all communication the data is sent sequential from one node to the next, down the line, while all the other nodes wait. Thus, the time required for a one-to-all communication is the time required for the data to travel all the way down the line of nodes. However, during all-to-all communication, data is cycled down the line from every node. Thus, every node is always busy, but the total communication time is still only the time it takes for data to travel once down the line.

In the interest of minimizing data transfer, all the ensemble output perturbation vectors $\delta\hat{\mathbf{y}}^j_{k|k-1}$ are thus transferred to every node, where $L^{(2)}_k$ can be computed locally. This requires only one all-to-all communication call for the ensemble output perturbation vectors. Conversely, if the summation components of $L^{(2)}_k$ were computed locally, an all-to-all communication of the entire matrix would be necessary, increasing communication significantly while decreasing computation only slightly.

In the EnKF framework, each individual ensemble member is assimilated with a noisy measurement. The noisy measurement on process p^j is denoted $\mathbf{d}^j_k$ and is found by adding random noise on top of the original measurement (from the truth model), with statistics consistent with the known properties of the sensors:

$$\mathbf{d}^j_k = \mathbf{y}_k + \mathbf{v}^j_k. \quad (70)$$

The statistics of the added noise mirror the known measurement noise of (1b). This gives the forcing to each ensemble member estimate $\mathbf{z}^j_k$ as

$$\mathbf{z}^j_k = \mathbf{d}^j_k - H \hat{\mathbf{x}}^j_{k|k-1} = \mathbf{y}_k + \mathbf{v}^j_k - \hat{\mathbf{y}}^j_{k|k-1}, \quad (71)$$

where $\hat{\mathbf{y}}^j_{k|k-1}$ is the ensemble output vector on each process. Hence, the calculation of this vector can be done locally; no message passing is required, other than to provide each process with the truth model measurement $\mathbf{y}_k$.

At this point, a simple linear system needs to be solved [due to the $\left(L^{(2)}_k\right)^{-1}$ term] on each process. This solve is straightforward because $L^{(2)}_k \in \Re^{m\times m}$ is both symmetric and relatively small. Many algorithms exist for the efficient solution of such systems.

Note that, with the assumption $\mathcal{R} > 0$, the matrix $L_k^{(2)}$ is, in general, nonsingular, and thus the solution to the following system exists and is unique:

$$\mathbf{u}_k^j = \left(L_k^{(2)}\right)^{-1} \mathbf{z}_k^j \Rightarrow \; L_k^{(2)}\,\mathbf{u}_k^j = \mathbf{z}_k^j. \tag{72}$$

With the computation of $\mathbf{u}_k^j$ done locally on each process, the update equation (63) can again be rewritten as:

$$\hat{\mathbf{x}}_{k|k}^j = \hat{\mathbf{x}}_{k|k-1}^j + L_k^{(1)}\,\mathbf{u}_k^j. \tag{73}$$

Substituting in the definition of $L_k^{(1)}$ from (68), this update becomes:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k}^j &= \hat{\mathbf{x}}_{k|k-1}^j + \left[\frac{1}{N-1}\sum_{i=1}^{N} \delta\hat{\mathbf{x}}_{k|k-1}^i \,(\,\delta\hat{\mathbf{y}}_{k|k-1}^i\,)^H\right]\mathbf{u}_k^j \\ &= \hat{\mathbf{x}}_{k|k-1}^j + \sum_{i=1}^{N}\left[\frac{(\,\delta\hat{\mathbf{y}}_{k|k-1}^i\,)^H\,\mathbf{u}_k^j}{N-1}\right]\delta\hat{\mathbf{x}}_{k|k-1}^i, \\ \Rightarrow \;\; \hat{\mathbf{x}}_{k|k}^j &= \hat{\mathbf{x}}_{k|k-1}^j + \sum_{i=1}^{N} \gamma_k^{ij}\,\delta\hat{\mathbf{x}}_{k|k-1}^i \end{aligned} \tag{74a}$$

$$\text{where} \quad \gamma_k^{ij} = \frac{(\,\delta\hat{\mathbf{y}}_{k|k-1}^i\,)^H\,\mathbf{u}_k^j}{N-1}. \tag{74b}$$

In its final form, the measurement update equation (74) updates each ensemble member via a linear combination of each ensemble state perturbation vector $\delta\hat{\mathbf{x}}_{k|k-1}^j$. This form eliminates the need for any additional storage arrays. The update can be computed in an all-to-all round robin format, where the ensemble state perturbation vector on each process is shifted one hop to the adjacent process. Then, the corresponding update is computed on every process, and the data is shifted again. Overall, the total communication is equivalent to a single all-to-all send of the ensemble state perturbation vector, but because the computation is done in between each message hop, there is no accumulating storage necessary.

In this manner, both the forward EnKF updates (8) and the retrograde EnKF "downdates" (51) can be computed numerically, even in large-scale systems. Left to compute are the adjoint marches. These can be done in a similar manner as traditional 4DVar techniques, with the exception of the additional storage/checkpointing required by 4DVar but not required by EnVE, as discussed near the end of Section 3.

At the completion of the adjoint march, the gradient is calculated from the adjoint ensemble at t_{-K} and the deviation from the background term. For the background component of the cost function, the pseudoinverse of the ensemble background covariance $(\mathcal{P}_{-K|-K}^e)^+$ must be computed. In general, this is computationally intensive, but here the intrinsic structure of the ensemble framework can again be exploited to simplify this calculation. For clarity, the time subscripts will be dropped. Recall from (9) that:

$$\mathcal{P}^e = \frac{(\delta\widehat{X})\,(\delta\widehat{X})^H}{N-1}, \;\text{ where }\; \delta\widehat{X} = \left[\delta\hat{\mathbf{x}}^1 \quad \delta\hat{\mathbf{x}}^2 \quad \cdots \quad \delta\hat{\mathbf{x}}^N\right],$$
$$\delta\hat{\mathbf{x}}^j = \hat{\mathbf{x}}^j - \overline{\mathbf{x}}, \quad \text{and} \quad \overline{\mathbf{x}} = \frac{1}{N}\sum_j \hat{\mathbf{x}}^j.$$

Define the reduced singular value decomposition (SVD) of $\delta\widehat{X}$ as

$$\delta\widehat{X} = U\,\Sigma\,V^H. \tag{75}$$

Note that, though $\widehat{X}$ is assumed to have full column rank (i.e., the ensemble members are assumed to be linearly independent), the process of determining the perturbations of these ensemble members reduces the rank of $\delta\widehat{X}$ by one (due to the subtraction of the ensemble mean). Therefore, the reduced singular value decomposition results in $N-1$ singular values σ_i that make up the diagonal of the $(N-1)\times(N-1)$ matrix Σ. Using the reduced SVD of $\delta\widehat{X}$, the background ensemble covariance can be expressed as:

$$\mathcal{P}^e = \frac{(U\,\Sigma\,V^H)\,(U\,\Sigma\,V^H)^H}{N-1} \tag{76}$$

$$= \frac{U\,\Sigma\,V^H\,V\,\Sigma\,U^H}{N-1}, \tag{77}$$

$$\Rightarrow \;\; \mathcal{P}^e = \frac{U\,\Sigma^2\,U^H}{N-1}, \tag{78}$$

where $V^H\,V = I$. With $\mathcal{P}^e$ in this SVD form, the pseudoinverse is recognized immediately as

$$(\mathcal{P}^e)^+ = (N-1)\,U\,\Sigma^{-2}\,U^H. \tag{79}$$

Thus, finding $(\mathcal{P}^e)^+$ reduces to the problem of finding an orthonormal basis for the column space of $\delta\widehat{X}$. To do this, recall that $\delta\widehat{X} \in \Re^{n\times N}$ where $N \ll n$. Thus, it is more efficient to find first an orthonormal basis for the row space of $\delta\widehat{X}$. This is done via an eigendecomposition of the following matrix:

$$(\delta\widehat{X})^H\,(\delta\widehat{X}) = \begin{bmatrix} V & \mathbf{v}\end{bmatrix}\begin{bmatrix}\Sigma^2 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}V^H\\ \mathbf{v}^H\end{bmatrix}, \tag{80a}$$

$$\Rightarrow \qquad V^H = W = \begin{bmatrix}\mathbf{w}_1 & \cdots & \mathbf{w}_N\end{bmatrix}, \tag{80b}$$

$$\Sigma^2 = \text{diag}(\,\sigma_1^2, \sigma_2^2, \cdots, \sigma_{N-1}^2\,), \tag{80c}$$

$$\sigma_1 \;\geqslant\; \sigma_2 \;\geqslant\; \cdots \;\geqslant\; \sigma_{N-1} \;>\; 0, \tag{80d}$$

where $\mathbf{v}$ is a vector representing the basis for the null space of $(\delta\widehat{X})^H\,(\delta\widehat{X})$, due to the fact that $\delta\widehat{X}$ is made rank deficient by subtracting off its mean. Each σ_i^2 is guaranteed real and positive by construction. As N is the number of ensemble members (on the order of 10^2), many efficient algorithms exist for the computation of this spectral decomposition. Once the row space V has been found, it is easily shown from (75) that the column space U is given by

$$U = \delta\widehat{X}\,V\,\Sigma^{-1} \tag{81}$$

which, leveraging (80a), can be substituted directly back into (79) to give:

$$(\mathcal{P}^e)^+ = (N-1)\,\delta\widehat{X}\,V\,\Sigma^{-4}\,V^H\,\delta\widehat{X}^H. \tag{82}$$

Now, by defining the following vector

$$\mathbf{s}^T = \begin{bmatrix}1/\sigma_1^4 & \cdots & 1/\sigma_{N-1}^4\end{bmatrix}, \tag{83}$$

the pseudoinverse (82) can be rewritten as:

$$(\mathcal{P}^e)^+ = (N-1)\left(\sum_{i=1}^{N}\delta\hat{\mathbf{x}}^i\,(\mathbf{w}_i \bullet \mathbf{s})^H\right)\left(\sum_{j=1}^{N}\mathbf{w}_j\,(\delta\hat{\mathbf{x}}^j)^H\right), \tag{84}$$

where $\mathbf{a}\bullet\mathbf{b}$ denotes the Schur product (element-wise multiplication) of the corresponding vectors. In practice, this matrix will never be explicitly computed. Rather, this matrix is always used as a part of a matrix/vector product of the form

$$(\mathcal{P}^e)^+\mathbf{z} = (N-1)\left(\sum_{i=1}^{N}\delta\hat{\mathbf{x}}^i\,(\mathbf{w}_i \bullet \mathbf{s})^H\right)\left(\sum_{j=1}^{N}\mathbf{w}_j\,(\delta\hat{\mathbf{x}}^j)^H\right)\mathbf{z}. \tag{85}$$

Now noting that the general vector $\mathbf{z}$ does not depend on j, it can be brought inside the second summation, giving an inner product (that results in a scalar) as follows:

$$(\mathcal{P}^e)^+\mathbf{z} = (N-1)\left(\sum_{i=1}^{N}\delta\hat{\mathbf{x}}^i\,(\mathbf{w}_i \bullet \mathbf{s})^H\right)\left(\sum_{j=1}^{N}(\mathbf{z}^H\delta\hat{\mathbf{x}}^j)\,\mathbf{w}_j\right). \tag{86}$$

Lastly, the second summation in (86) results in another vector (this time independent of i) that can be brought inside the first summation to give the final form implemented numerically:

$$(\mathcal{P}^e)^+\mathbf{z} = \sum_{i=1}^{N} \beta_i\, \delta\hat{\mathbf{x}}^i, \quad \beta_i = (N-1)\sum_{j=1}^{N} (\mathbf{z}^H \delta\hat{\mathbf{x}}^j)(\mathbf{w}_i \bullet \mathbf{s})^H \mathbf{w}_j. \tag{87}$$

Hence, the product of the ensemble covariance pseudoinverse $(\mathcal{P}^e)^+$ with any general vector $\mathbf{z}$ is shown to be a simple linear combination of the ensemble perturbation vectors $\delta\hat{\mathbf{x}}^j$, as was the case with the ensemble update formula (74). Unlike the update formula (74), the weighting β_i on each ensemble member perturbation requires a sum over the whole of the ensemble. This sum is trivial, though, as it requires simple all-to-all communication of the scalar $\mathbf{z}^H \delta\hat{\mathbf{x}}^j$. It is assumed here that the eigenvector matrix V and the singular values σ have been computed in parallel on each ensemble, and thus no additional communication is necessary to compute this portion of the scalar sum. Then, to leading order, for each variational iteration, the pseudoinverse computations require only one all-to-all send (of the ensemble state perturbations) to build up the matrix $(\delta\widehat{X})^H\ (\delta\widehat{X})$, prior to the spectral decomposition, and one round-robin all-to-all send (again, of the ensemble state perturbations) for each matrix/vector product computation. No notable extra storage is required for these computations.

5 EnVE Consistency

At a specific time, given a linear system, a background estimate with known covariance, and a new measurement with known noise characteristics, the Kalman estimate is the best linear unbiased estimate (BLUE) that balances these two uncertainties to minimize a corresponding cost function. For linear systems, it is straightforward to think of the estimate at any time as being fully conditioned on a subset of measurements; thus the notation $\bar{\mathbf{x}}_{i|k}$. It is important to note that, even in the case in which the entire state is being measured, the optimal estimate is not simply the value of the observation at that time. Thus the importance of the background estimate, as it gives the existing estimate some "inertia", avoiding spurious updates due to outlying observations.

For LQG systems, one sequential forward march through a set of observations gives the optimal estimate, $\bar{\mathbf{x}}_{0|0}$, at the present time t_0. It is possible to smooth past estimates, say at t_{-k}, by marching the current estimate backward and retaining the information gained from all observations, giving the smoothed estimate $\bar{\mathbf{x}}_{-k|0}$. This smoothing march effectively conditions the past estimates on the now known future observations $\{\mathbf{y}_{1-k}\cdots\mathbf{y}_0\}$. It does not change the estimate based on any information from the past observations $\{\mathbf{y}_j \,|\, j<-k\}$, as this information has already been included. Now, given this smoothed estimate $\bar{\mathbf{x}}_{-k|0}$ and its associated covariance $\mathcal{P}_{-k|0}$, one could mistakenly run a KF forward again through the set of measurements $\{\mathbf{y}_{1-k}\cdots\mathbf{y}_0\}$. This would be an attempt to recondition the estimate on these measurements, and completely violates the optimality of the estimate. In fact, it is easy to show that such an approach, applied iteratively, would lead to an estimate that converges to the observations themselves, independent of the original background terms. This is exactly the type of inconsistency that EnVE has been constructed carefully to avoid. In the linear setting, it may seem obvious that a single observation must be used only once, but, in a nonlinear setting, where suboptimal sequential updates are performed and variational iterations are not taken completely to convergence, this issue becomes less obvious, and must be handled with care.

To achieve consistency (that is, to ensure that the answer given by the EnVE algorithm reduces to that given by the KF when the system is linear and the ensemble sufficiently large), EnVE must rigorously keep track of the background estimate. Ultimately, sequential methods (EnKF) and variational methods (4DVar) are used to solve the same problem. Both methods work to minimize a cost function to optimize the estimate at t_0 conditioned on all available measurements. Thus, when these cost functions are defined appropriately, it is possible to switch back and forth between sequential and variational methods consistently, as EnVE does. For a linear system with a set of measurements defined on $[t_{-K}, t_0]$, the smoothed KF estimate at t_{-K}, $\bar{\mathbf{x}}_{-K|0}$ (found by marching a KF forward through the observations and marching the resulting estimate backward to t_{-K}), is identical to the solution of a converged 4DVar algorithm with an appropriately defined background term. In other words, the optimal smoothed KF estimate $\bar{\mathbf{x}}_{-K|0}$ is the global minimum of the 4DVar cost function in the case of a linear system. For nonlinear systems, the optimal estimate at t_{-K} can not be found directly via a sequential algorithm in this manner, though the smoothed KF estimate is indeed an appropriate initial guess for an iterative (variational) algorithm.

This relationship is what EnVE attempts to exploit to improve the estimate. Marching an Ensemble Kalman Smoother (EnKS) will not produce the optimal smoothed estimate $\bar{\mathbf{x}}_{-K|0}$ because of the nonlinearities in the system and the approximations required for the ensemble framework. However, by removing the effect of the measurements and appropriately defining the 4DVar cost function background term, this sub-optimal smoothed estimate can be used as an initial condition for the variational step. If the smoothed estimate $\bar{\mathbf{x}}_{-K|0}$ happens to be optimal (that is, if the system considered is essentially linear), then the variational iteration is already converged and will produce a zero update to the estimate. Thus, EnVE uses the EnKS to initialize the 4DVar optimization, but does not reuse the information in the observations inconsistently. EnVE therefore reduces to the expected optimal results of the Kalman Smoother (KS) for a linear system.

A cartoon of the expected estimation error as EnVE progresses for a typical chaotic system is shown in Figure 7. Due to the chaotic nature of the system, any forward march of an estimate will lead to exponential growth of the expected estimation error (shown linearly in semi-log coordinates). Each EnKF measurement update creates a discrete drop in the expected estimation error. When a variational iteration is performed, the estimate is marched backward. This causes an exponential decrease in the expected error as trajectories of the chaotic system converge (along the attractor) during the backward march. Then, a variational update is made, further reducing the expected error, and the resulting estimate is propagated forward again to the next available measurement. Recall that with a linear system, the update due to the variational step will have zero length, thus returning the estimate back to its original state to continue the sequential march. This helps illustrate the consistent nature of EnVE.

6 Advantages

By combining the statistical capabilities of the EnKF along with the batch processing/smoothing capabilities of a variational method, EnVE builds a better estimate of the system at a justifiable computational cost. Using the EnKF to initialize a 4DVar-like

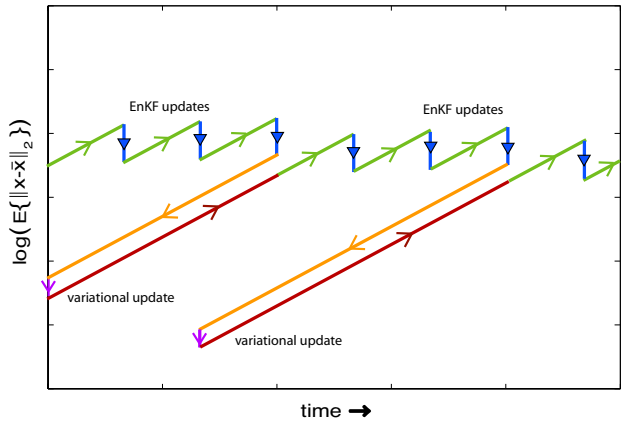

Figure 7. A cartoon illustrating the expected error for EnVE performed on a chaotic system. Exponential growth (linear growth in semi-log coordinates) in the expected error occurs during forward marches. Discrete reduction in the expected error occurs at both the sequential updates and the variational updates. Note that, with a linear system, the variational updates are necessarily zero, returning the estimates to their original values upon completion of the variational steps.

iteration allows for fewer variational steps to be used, because the initial condition for the variational iteration is much more accurate than the background estimate alone, and full convergence is not required. The intrinsic ability of the EnKF to represent the statistical properties of the estimate allows EnVE to repeatedly and consistently revisit past measurements and update the central trajectory of the ensemble (about which the system can be linearized when considering its covariance evolution) based on new measurements.

Two objectives in the development of EnVE were a multiscale-in-time analysis and a receding horizon optimization framework. The significance of these properties are highlighted in the following two subsections. Combined, these two properties create a dynamic optimization surface that tends to have desirable convergence properties for complex nonlinear systems.

6.1 Multiscale in Time

Because the variational window in EnVE is defined from the right (present time) by marching the current estimate backward until divergence, the width of this window can be selected during the iteration. In contrast, with traditional 4DVar, this window width must be specified in advance. The variable variational window width of EnVE can be used to precondition the optimization problem appropriately by coordinating this width with the accuracy of the initial estimate, as discussed previously and illustrated graphically in Figure 8.

Due to the noise in the measurements, a short window containing only a few observations is prone to inaccuracy. That is, the global minimum of the cost function defined over only a few observations is likely to deviate significantly from the "truth". However, because only a few measurements are included in this short window (with corresponding short marches of the chaotic system), this optimization surface tends to be fairly regular, with a large region of attraction to the global minimum. The size of the region of attraction is important when using gradient-based algorithms, as such algorithms are prone to converge to local minima.

As the estimate improves, longer windows with more included observations are utilized by EnVE. This tends to make the optimization surface more irregular, and to shrink the region of attraction to the global minimum. Thus, this extension of the variational window needs to be done gradually enough that the improved estimate remains in this reduced region of attraction. Because more measurements are included in such longer windows, the effect of sensor noise is diminished (as compared to the shorter windows), making the global minimum more accurate with respect to the "truth" as the window length is increased.

6.1.1 Example: Multiscale Preconditioning of a 1D Optimization

To further understand the effect of varying the variational window width on the optimization surface and convergence, consider the (cartoonish) 1D example indicated in Figure 9. The toy system considered is an estimation problem based on a Lorenz system (see Section 7) in which two of the three components of the initial state, $x_1(0)$ and $x_3(0)$, are assumed to be known; however, the precise details of this toy system are relatively unimportant for the purpose of the present discussion. For the purpose of illustration, discrete noisy measurements are taken at a constant sampling rate and then smoothed to create a continuous-time measurement signal $y(t)$.

A simple cost function in then defined as the misfit between the measurement signal $y(t)$ and the evolution of the nonlinear system:

$$\mathcal{J}(u) = \int_0^T \| y(t) - x_2(t) \|_2 \, dt. \tag{88}$$

This cost function is a function of the initial condition $u = x_2(0)$ at the left edge of the window (here renormalized to be $t = 0$), and is parameterized by the width of the variational window T. For a given window width T, the estimate u at the left edge of the window is varied to determine the complete optimization surface of this toy system, the global minimum of this optimization surface, the distance of this global minimum from the "truth" (that is, distance of the global minimum from the value of $x_2(0)$ in the "truth" simulation used to generate the measurements), and the region of attraction to the global minimum (assuming that a gradient-based search algorithm is to be used to find it). This global minimum is tracked in Figure 10, along with the upper and lower bounds of the region of attraction, as a function of the window width T. It is seen that the global minimum converges to the optimal "truth" model, as expected, as T is increased. However, the curves outlining the region of attraction to this global minimum are important to understand and appreciate. For small windows ($T < 0.5$), due to the lack of complexity in this 1D example, the optimization surface is convex. Thus, any initial condition will converge to the global minimum. For longer windows ($T > 0.5$) the region of attraction shrinks, requiring increasing precision of the initial estimate. This is especially true for the upper bound, where even a slight error will cause erroneous convergence to a local minimum.

To clarify this effect even further, Figure 11 shows two optimization surfaces for particular fixed T. In the top subfigure ($T = 0.5$), the surface is just beginning to lose its convexity. In the bottom subfigure ($T = 1.0$), the optimization surface has a very accurate global minimum, but it is clear here, with such a small region of attraction, how initial estimates with too much error could easily converge to poor local minima.

In a typical high-dimensional chaotic system, the optimization surfaces will necessarily be much more complicated, but the trends (with respect to the accuracy of the global minimum and the re-

[h!]

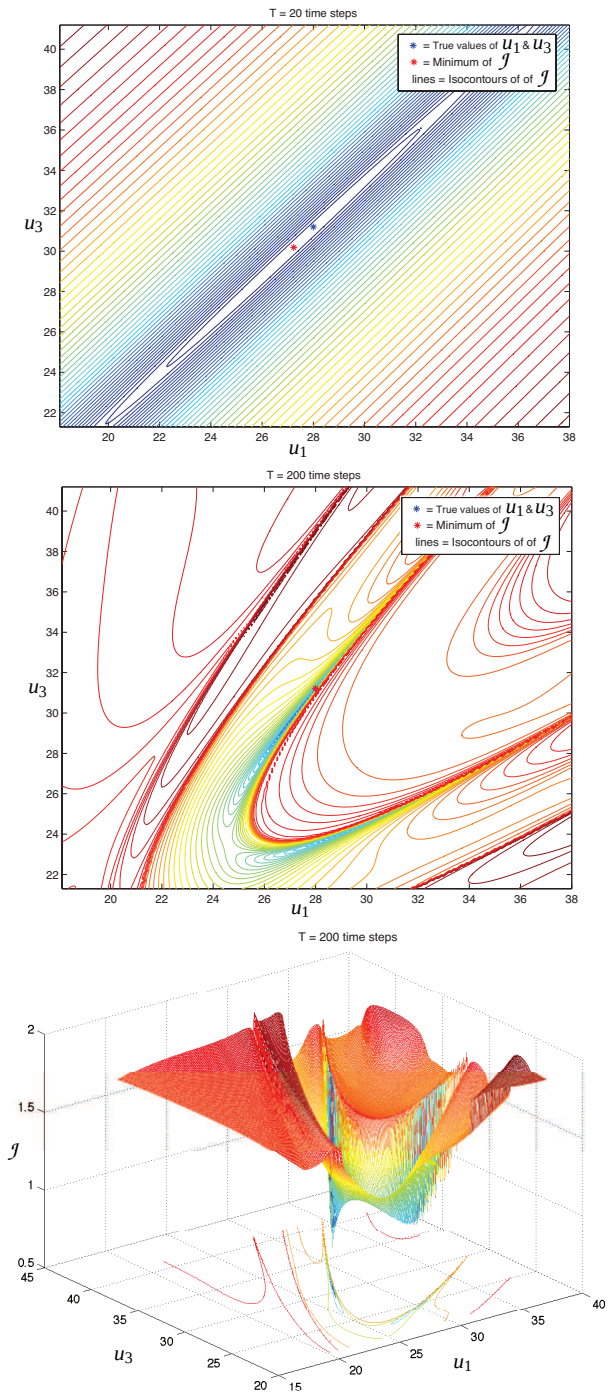

Figure 8. Representative plots illustrating the change in complexity of the optimization surfaces for a short ($T = 20$) variational window (*top*) and a long ($T = 200$) variational window (*middle* and *bottom*) for a test estimation problem related to the Lorenz equation (Section 7). Also shown is the known global minimum of the truth model, which is much closer to the global minimum of the highly irregular optimization surface of the longer window than it is to the minimum of the smoother surface of the shorter window.

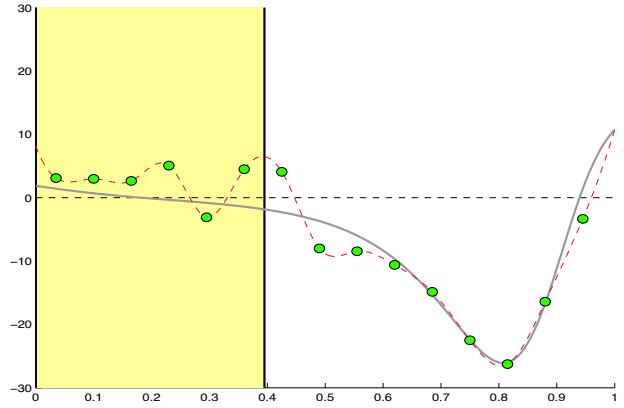

Figure 9. The second state, $x_2(t)$, of a Lorenz model (gray), and an (artificially perturbed) measurement signal generated from noisy measurements of this state (green dots). Given these measurements, and (for demonstration purposes only) knowledge of $x_1(0)$ and $x_3(0)$, we consider in Section 6.1.1 the scalar optimization problem of finding $u = x_2(0)$ in order to reconcile the trajectory of the estimate with the measurements over horizons of various widths T.

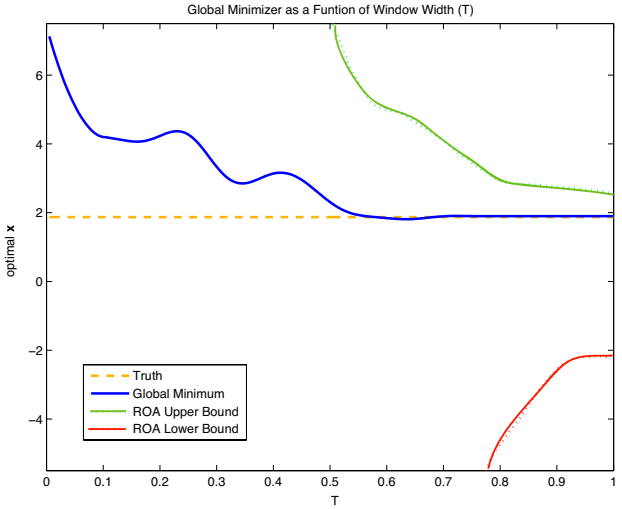

Figure 10. The global minimum (blue) of the cost function $J(u)$, plotted as a function of the window width T used to define the cost function. As the T increases, so does the proximity of this global minimum to the truth (dashed); however, the region of attraction to this global minimum (between the red and green curves) is also greatly reduced.

gion of attraction) are consistent with this 1D example. Thus, it is clear how a strategy that uses short variational windows for poor estimates and longer windows to further refine accurate estimates is indeed well founded.

6.2 *Receding Horizon*

A receding-horizon approach is defined by nudging the variational window forward in time to incorporate the most recent measurements obtained during each step of a variational optimization. Simplistic approaches to variational data assimilation leave the optimization window fixed until convergence. In contrast, EnVE redefines the optimization problem slightly at each iteration, updating it to include the newly-obtained measurements. As this modification causes the optimization surface to constantly shift, the algorithm

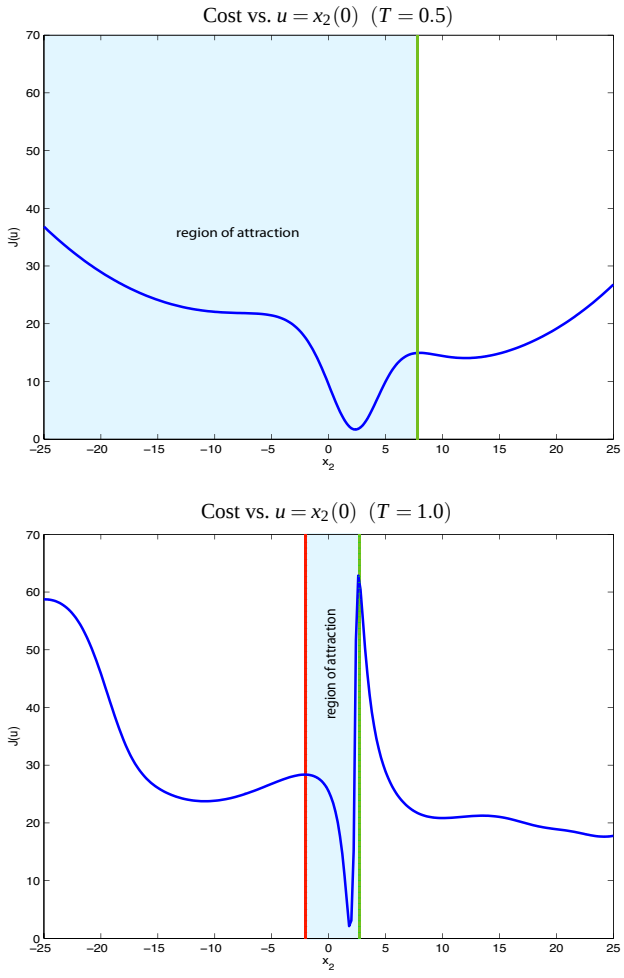

Figure 11. Optimization surfaces are shown for two variational window widths. Note the increase in complexity, even for a simple 1D problem, as this window width is increased. The lower plot indicates that the global minimum is very accurate, but that traditional gradient-based algorithms will only converge to this global minimum if the initial estimate is within a small region of attraction. This motivates the idea of increasing the window width gradually as convergence is approached.

never completely converges. However, the receding-horizon optimization framework updates the current estimate at each iteration with maximal efficiency, as it is constantly using the most up-to-date information available. Further, the resulting dynamic evolution of the optimization surface in fact helps to nudge the estimate out of the local minima into which it might otherwise settle.

A typical contrast between two forecasts [one generated with a fixed-horizon 4DVar algorithm and the other with the receding-horizon EnVE algorithm] is shown in Figures 12-13. Unlike EnVE, due to the computation time required for convergence of the fixed-horizon 4DVar algorithm, the corresponding variational window over which the optimization was performed has slipped far into the past. Due to the chaotic nature of the system of interest, any forecast diverges exponentially when marched into the future. Consequently, much of the relevant range of the fixed-horizon 4DVar forecast is wasted predicting events that have in fact already taken place. EnVE avoids this effect by keeping the variational window current, updating it at every iteration.

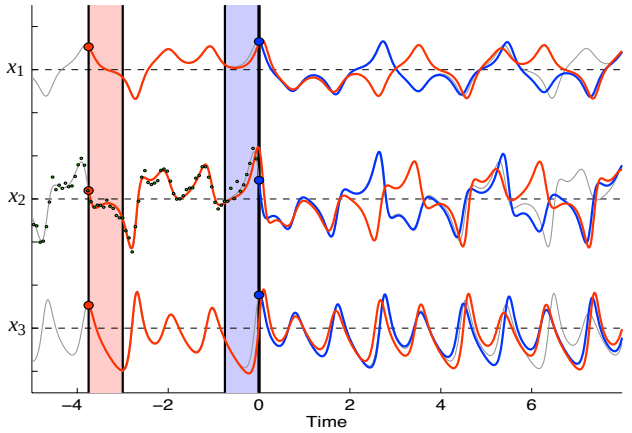

Figure 12. Two forecasts of a Lorenz system (light gray) with noisy measurements (small black dots). The red forecast is from a converged estimate of a fixed-horizon 4DVar algorithm, where the variational window considered has shifted far into the past during the time spent completing the computational iterations required to solve the optimization problem. The blue forecast is from an estimate computed using the receding-horizon EnVE framework. The fixed-horizon 4DVar forecast (red) visibly diverges from the truth (light grey, underneath the blue curve for much of the plot) near $t = 2$; the receding-horizon EnVE forecast (blue) visibly diverges from the truth (light grey) near $t = 6$.

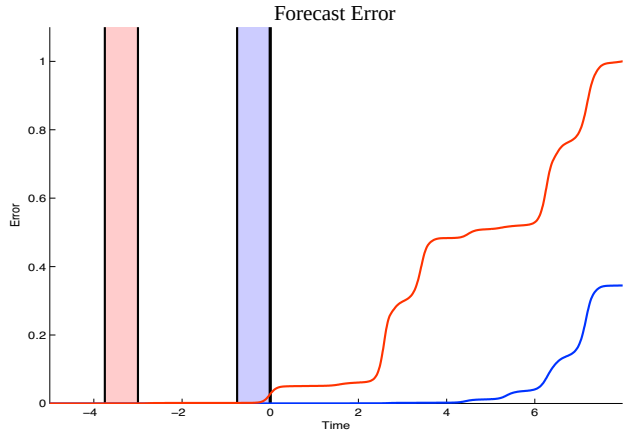

Figure 13. The integral in time of the square of the forecast errors from Figure 12 are shown. Note the difference in the accumulated errors of each of forecast is due in large part to the time the forecast is ahead of the latest variational window used during its optimization. As this time is significantly reduced in the receding-horizon framework, forecasts made a certain amount of time into the future are greatly improved.

6.3 *Parallel State/Adjoint Marches*

As already mentioned, another advantage of posing the variational optimization problem in a retrograde setting deals with the numerical implementation of EnVE. The adjoint equation is marched backward in time (from t_0 to t_{-K}), forced using the trajectory $\tilde{\mathbf{x}}(t)$. Typically, this trajectory is found by marching the initial condition $\tilde{\mathbf{x}}_{-K} = \mathbf{u}$ forward through the window (from t_{-K} to t_0). Especially for the multiscale systems of interest, this poses a large storage constraint on the problem, because the adjoint is forced by the whole trajectory, but in reverse order. In other words, the trajectory of $\tilde{\mathbf{x}}(t)$ needs to be computed and saved over the entire interval before the adjoint march can begin. Attempts to circumvent

this problem for large atmospheric-scale systems include the checkpointing algorithm, in which the trajectory is stored only on coarse time grid points, and then, as necessary, is either recomputed or linearly interpolated onto the fine (in time) grid used for timestepping the adjoint calculation. Checkpointing requires a substantial amount of storage and significantly increases the computation required to compute the adjoint.

Note that, with EnVE, this required estimate trajectory is determined backward in time rather than forward in time. Thus, the corresponding adjoint may be computed simultaneously, eliminating this storage problem altogether.

7 EnVE Test Case: Lorenz

As a simple first test case, the EnVE algorithm is implemented on the chaotic Lorenz system, first introduced by Lorenz (1963). For the Lorenz system, a three-dimensional ODE model is used with very noisy measurements of only the second state. Figure 14 represents a time history of all three states with the truth model shown in grey, the forecast shown in blue, and the present time represented by the thick, vertical, black line. The yellow box shows the variational window that is currently being revisited. Even with such a simple system, the results of the EnVE algorithm are very promising. This example illustrates the benefit of the multiscale variational window combined with the receding horizon framework to produce an accurate estimate of the present time.

8 Summary and Conclusions

In this paper, a new hybrid data assimilation method is introduced: Ensemble Variational Estimation (EnVE). The new method leverages the nonlinear statistical propagation properties of the sequential EnKF/EnKS to initialize and properly define an appropriate variational iteration, similar to 4DVar. This variational iteration is posed in such a way as to allow for a multiscale-in-time, receding-horizon optimization framework. The smoothed estimate from the EnKF is used as an accurate initial condition for the variational iteration, thus improving its overall performance. A multiscale-in-time framework is achieved via a retrograde march of the current estimate over the available observations. This multiscale-in-time framework appropriately preconditions the variational step. It also allows for a concurrent, parallel march of the appropriate adjoint equation, which is forced by the backward march of the estimate. Thus, no additional storage is required for the gradient computation, in sharp contrast with the significant additional storage typically required by a 4DVar implementation. Because the variational window width is a function of the accuracy of the estimate, EnVE tends to update poor estimates with short windows and more accurate estimates with longer windows.

An EnVE implementation on a simple Lorenz system was considered as a first application. Current work is focused on implementing EnVE on more complicated chaotic PDE systems. Preliminary results show a definite improvement using EnVE (over either EnKF or 4DVar alone) for assimilating data related to a passive scalar release in a complex unsteady 2D flowfield.

In summary, EnVE is a convenient and consistent hybrid of the basic EnKF and 4DVar algorithms already in wide use. Much of the current work on the EnKF and 4DVar may be applied directly to the EnVE algorithm while maintaining EnVE's consistency and desirable numerical properties. With such combined efforts, it might well be possible to develop significantly improved large-scale data assimilation algorithms in the years to come.

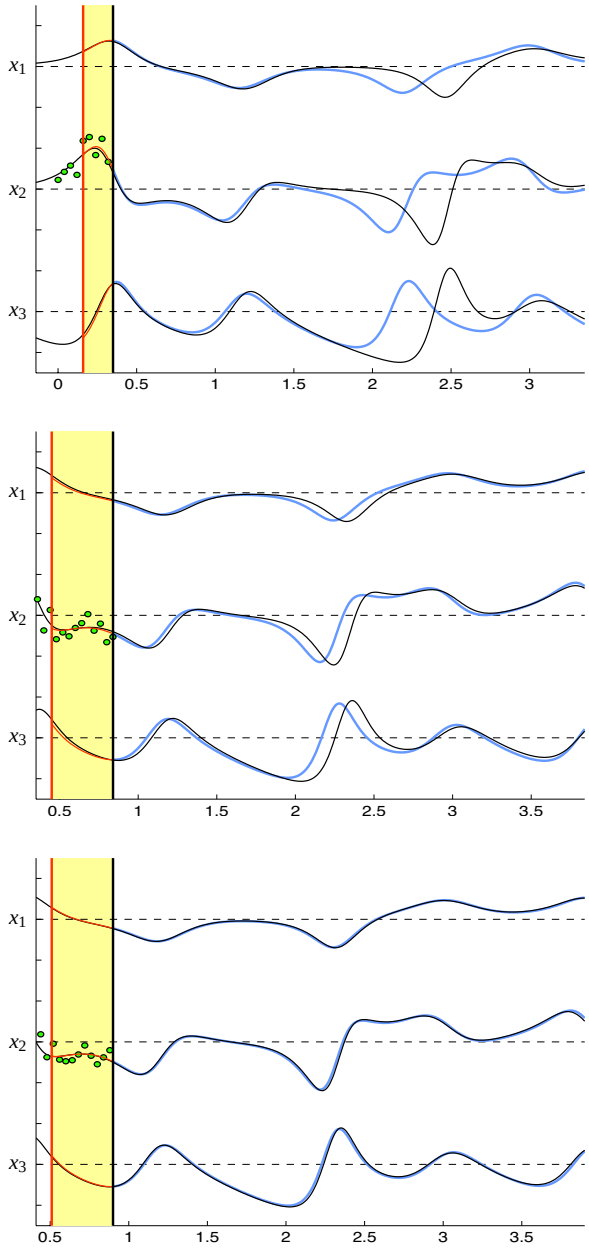

Figure 14. The EnVE algorithm demonstrated on a simple chaos model, the Lorenz system, with very noisy measurements (marked as green dots). (*top*) Initially, the estimate is fairly poor, as easily seen by the quickly diverging forecast (blue) from the truth model (black). The optimization window determined by EnVE for this iteration is fairly short, using only a few measurements to update the current estimate. (*center*) As the estimate is improved, the variational window selected by EnVE expands, helping to reduce further the error in the forecast. (*bottom*) Finally, with the expanded window, the estimate converges very accurately to the global minimum. At this point, the estimate tends to track the global minimum quite well. Occasionally, due to the chaotic nature of the system, the estimate may begin to diverge from the truth model. The spread of the ensemble indicates this increased uncertainty, and the EnVE algorithm responds by shortening the variational window used to again refine the estimate as quickly as possible.

9 Acknowledgments

The authors gratefully acknowledge the generous financial support of the National Security Education Center (NESC) at Los Alamos National Laboratory (LANL) and numerous helpful discussions with Dr. Frank Alexander (LANL), Prof. Daniel Tartakovsky (UCSD), and Sean Summers (UCSD).

REFERENCES

Anderson, B.D.O. and Moore, J.B. 1979. *Optimal Filtering*, Dover.

Anderson, J.L., 2001. An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review* **129**, 2884–2903.

Berre, L., Pannekoucke, O., Desroziers, G., Ştefănescu, S.E., Chapnik, B., and Raynaud, L. 2007. A variational assimilation ensemble and the spatial filtering of its error covariances: increase of sample size by local spatial averaging. *Proceedings of the ECMWF workshop on Flow-dependent aspects of Data Assimilation. Reading.* 151–168.

Bewley, T. 2008. *Numerical Renaissance: Simulation, Optimization, and Control*, Under preparation.

Bewley, T., Cessna, J., Colburn, C. and Zhang, D. 2008. EnVE: Hybird ensemble/variational estimation and its applications to multiscale chaotic PDE model systems. *Under preparation*.

Bewley, T., Cessna, J., Colburn, C. and Zhang, D. 2008. EnVO: An ensemble/variational observation strategy for minimizaton of forecast uncertainty. *Under preparation.*

Bishop, C.H., Etherton, B.J. and Majumdar, S.J. 2001. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly Weather Review* **129**, 420–436.

Bouttier, F. and Courtier, P. 2002. Data assimilation concepts and methods, March 1999. *Meteorological training course lecture series. ECMWF*.

Caya, A., Sun, J. and Snyder, C. 2005. A comparison between the 4DVar and the ensemble Kalman filter techniques for radar data assimilation. *Monthly Weather Review* **133**, 3081–3094.

Cessna, J., Colburn, C. and Bewley, T. 2007. Multiscale retrograde estimation and forecasting of chaotic nonlinear systems. *Proceedings from 46th IEEE Conference on Decision and Control*, New Orleans.

Cohn, S.E., Sivakumaran, N.S. and Todling, R. 1994. A fixed-lag Kalman smoother for retrospective data assimilation. *Monthly Weather Review* **122**, 2838–2867.

Colburn, C. and Bewley, T. 2008. An efficient reversible pseudo-random number generator. *Under preaparation*.

Evensen, G., 1994. Sequential data assimilation with a non-linear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Review* **99**, 10143–10162.

Evensen, G., 2003. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics* **53**, 343–367.

Evensen, G. and van Leeuwen, P.J. 2000. An ensemble Kalman smoother for nonlinear dynamics. *Monthly Weather Review* **128**, 1852–1867.

Fisher, M. 2002. Assimilation Techniques (4): 4dVar, April 2001. *Meteorological training course lecture series. ECMWF*.

Ghil, M., Cohn, S., Tavantzis, J., Bube, K. and Isaacson, E 1981. Applications of estimation theory to numerical weather prediction. *Dynamic meteorology, data assimilation methods*, 139–224.

Gustaffson, N. 2007. Discussion on '4D-Var or EnKF?' *Tellus* **59A**, 774–777.

Hamill, T.M. and Snyder, C. 2000. A hybrid ensemble Kalman filter-3D variational analysis scheme. *Monthly Weather Review* **128**, 2905–2919.

Hamill, T.M., Whitaker, J.S. and Snyder, C. 2001. Distance-dependent filtering of the background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review* **129**, 2776–2790.

Hoteit, I., Pham, D.-T., Triantafyllou, G., and Korres, G. 2008. Particle Kalman Filtering for Data Assimilation in Meteorology and Oceanography. *Under Preparation*.

Houtekamer, P.L. and Mitchell, H.L. 1998. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review* **126**, 796.

Houtekamer, P.L. and Mitchell, H.L. 2001. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review* **129**, 123–137.

Hunt, B.R., Kalnay, E., Kostelich, E.J., Ott, E., Patil, D.J., Sauer, T., Szunyogh, I., Yorke, J.A. and Zimin, A.V. 2004. Four-dimensional ensemble Kalman filtering. *Tellus* **56A**, 273–277.

Kailath, T., 1973. Some new algorithms for recursive estimation in constant linear systems. *IEEE Transactions on Information Theory* **19**, 750–760.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* **82**, 35–45.

Kalman, R.E. and Bucy, R.S. 1961. New results in linear filtering and prediction theory. *Journal of Basic Engineering* **83**, 95–108.

Kalnay, E., Li, H., Miyoshi, T., Yang, S.-C., and Ballabrera-Poy, J. 2007. 4D-Var or Ensemble Kalman Filter? *Tellus* **59A**, 758–773.

Kim, J. and Bewley, T. 2007. A linear systems approach to flow control. *Annual Review of Fluid Mechanics* **39**, 383–417.

Kim, S., Eyink, G.L., Restrepo, J.M., Alexander, F.J. and Johnson, G. 2003. Ensemble filtering for nonlinear dynamics. *Monthly Weather Review* **131**, 2586–2594.

Kraus, T., Kühl, P., Wirsching, L., Bock, H.G. and Diehl, M. 2006. A Moving Horizon State Estimation algorithm applied to the Tennessee Eastman Benchmark Process. *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* 377–382.

Lattès, R. and Lions, J.L. 1969. *The method of quasi-reversibility: applications to partial differential equations*. American Elsevier, New York.

Le Dimet, F.-X. and Talagrand, O. 1986. Variational algorithms for analysis and assimilation of meteorological observations; theoretical aspects. *Tellus* **38A**, 97.

Li, Z. and Navon, I.M. 2001. Optimality of 4D-Var and its relationship with the Kalman filter and the Kalman smoother. *Quarterly Journal of the Royal Meteorological Society* **127**, 661–684.

Lorenc, A.C., 1986. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society* **112**, 1177–1194.

Lorenc, A.C., 2003. The potential of the ensemble Kalman filter for NWP-a comparison with 4D-Var. *Quarterly Journal of the Royal Meteorological Society* **129**, 3183–3203.

Lorenz, E.N., 1963. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences* **20**, 130–141.

Michalska, H. and Mayne, D.Q. 1995. Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control* **40**, 995–1006.

Ott, E., Hunt, B.R., Szunyogh, I., Zimin, A.V., Kostelich, E.J., Corazza, M., Kalnay, E., Patil, D.J. and Yorke, J.A. 2004. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus* **56A**, 415–428.

Parrish, D.F. and Derber, J.C. 1992. The National Meteorological Center's spectral statistical interpolation and analysis system. *Monthly Weather Review* **120**, 1747–1763.

Protas, B., Bewley, T.R. and Hagen, G. 2004. A computational framework for the regularization of adjoint analysis in multiscale PDE systems. *Journal of Computational Physics* **195**, 49–89.

Rabier, F., Thepaut, J.-N. and Courtier, P. 1998. Extended assimilation and forecast experiments with a four-dimensional variational assimilation system. *Quarterly Journal of the Royal Meteorological Society* **124**, 1861–1887.

Rauch, H.E., Tung, F. and Striebel, C.T. 1965. Maximum Likelihood Estimates of Linear Dynamic Systems. *AIAA Journal* **3**, 1445–1450.

Wan, E.A. and van der Merwe, R. 2000. The unscented Kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000* 153–158.

Whitaker, J.S. and Hamill, T.M. 2002. Ensemble data assimilation without perturbed observations. *Monthly Weather Review* **130**, 1913–1924.

Zhang, M, Zhang, F, and Hansen, J. 2007. Coupling ensemble Kalman filter with four-dimensional variational data assimilation. *22nd Conference on Weather Analysis and Forecasting/18th Conference on Numerical Weather Prediction, Park City.*