A comparison of the equivalent weights particle filter and the local ensemble transform Kalman filter in application to the barotropic vorticity equation

Tellus **000**, 000–000 (0000) Printed 19 September 2016 (Tellus LaTeX style file v2.2)

A comparison of the equivalent weights particle filter and the local ensemble transform Kalman filter in application to the barotropic vorticity equation

By P. A. Browne⋆, *Department of Meteorology, University of Reading, UK*

(Manuscript received xx xxxx xx; in final form xx xxxx xx)

ABSTRACT

Data assimilation methods that work in high dimensional systems are crucial to many areas of the geosciences: meteorology, oceanography, climate science etc. The equivalent weights particle filter has been designed, and has recently been shown to, scale to problems that are of use to these communities. This article performs a systematic comparison of the equivalent weights particle filter with the established and widely used local ensemble transform Kalman filter. Both methods are applied to the barotropic vorticity equation for different networks of observations. In all cases it was found that the local ensemble transform Kalman filter produced lower root mean squared errors than the equivalent weights particle filter. The performance of the equivalent weights particle filter is shown to depend strongly on the form of nudging used, and a nudging term based on the local ensemble transform Kalman smoother is shown to improve the performance of the filter. This indicates that the equivalent weights particle filter must be considered as a truly 2-stage filter and not only by its final step which avoids weight collapse.

Keywords: Equivalent weights particle filter, nonlinear data assimilation, EMPIRE, LETKF, nudging, LETKS relaxation

1. Introduction

1.1. Data assimilation and Bayes' theorem

When making a prediction based on a dynamical model of a system it is necessary to initialise that model. This could be done simply by guessing the initial conditions of such a model or, as is more common, confronting the model with observations of the system.

Such observations necessarily have errors associated with them and also tend to be incomplete. That is, they are not direct observations of every component of the model. The mathematical formulation of how to rigorously incorporate such observations into the model is Bayes' theorem (Bayes and Price, 1763; Jazwinski, 1970):

$$p(x \mid y) = \frac{p(x)p(y \mid x)}{p(y)}. \qquad (1)$$

In this equation x represents the model state and y the observations. Hence the posterior pdf $p(x \mid y)$ is given as the product of the likelihood $p(y \mid x)$ with the prior $p(x)$ and normalised by the pdf of the observations $p(y)$. Different approximations of Bayes' theorem lead to different methods of data assimilation. For instance if one reduces the problem to finding a local mode of the posterior pdf, this becomes an inverse problem which can be solved by variational techniques: the famous 3DVar and 4DVar methods (see for example Le Dimet and Talagrand (1986); Dashti et al. (2013)).

1.2. Particle filters

A particle filter is a Monte-Carlo approach to computing the posterior via Bayes' theorem (see for example Smith et al. (2013) or van Leeuwen (2009)) in the context of a dynamically evolving system.

Without loss of generality, suppose that we have the prior pdf, $p(x^k)$, at timestep k written as a finite sum of delta functions (formally distributions),

$$p(x^k) = \sum_{i=1}^{N_e} w_i^k \delta(x^k - x_i^k) \qquad (2)$$

where $\delta(x)$ is the standard Dirac delta function. The set of state vectors $x_i^k, i = 1, \ldots, N_e$ is known as the *ensemble* and each state vector is referred to interchangeably as a *particle* or *ensemble member*. Note that in this notation the prior is arbitrary:

⋆ Corresponding author.
e-mail: p.browne@reading.ac.uk

it may depend on any data that has previously been assimilated and may have been evolved from a known probability density at a previous time. This information will be encoded into the *weights* w_i^k. As $p(x^k)$ is a pdf, $\int p(x^k)\,\mathrm{d}x^k = 1$ which implies $\sum_{i=1}^{N_e} w_i^k = 1$ and $p(x^k) \geqslant 0$ implies $w_i^k \geqslant 0$.

We have a model for the dynamics of the state which is Markovian:

$$x^{k+1} = f(x^k) + \beta^k \tag{3}$$

where f is a deterministic model and β^k is a stochastic model error term. To evolve the prior in time we note that, from the definition of conditional probability,

$$p(x^{k+1}) = \int p(x^{k+1} \mid x^k) p(x^k)\,\mathrm{d}x^k. \tag{4}$$

Now following Gordon et al. (1993), $p(x^{k+1} \mid x^k)$ is a Markov model defined by the statistics of β^k that are assumed known:

$$p(x^{k+1} \mid x^k) = \int p(x^{k+1} \mid x^k, \beta^k) p(\beta^k \mid x^k)\,\mathrm{d}\beta^k. \tag{5}$$

As β^k is independent of the state x^k, $p(\beta^k \mid x^k) = p(\beta^k)$ and we have

$$p(x^{k+1} \mid x^k) = \int p(\beta^k)\delta\Big(x^{k+1} - [f(x^k) + \beta^k]\Big)\,\mathrm{d}\beta^k. \tag{6}$$

Substituting (2) and (6) into (4) we obtain

$$p(x^{k+1}) = \iint p(\beta^k)\delta\Big(x^{k+1} - [f(x^k) + \beta^k]\Big)\,\mathrm{d}\beta^k \sum_{i=1}^{N_e} w_i^k \delta(x^k - x_i^k)\,\mathrm{d}x^k. \tag{7}$$

Integrating over x^k this reduces to

$$p(x^{k+1}) = \sum_{i=1}^{N_e} w_i^k \int p(\beta^k)\delta\Big(x^{k+1} - [f(x_i^k) + \beta^k]\Big)\,\mathrm{d}\beta^k. \tag{8}$$

Now for each ensemble member i we make a single draw from $p(\beta^k)$, β_i^k (i.e. $p(\beta^k) = \delta(\beta^k - \beta_i^k)$) so that

$$\begin{aligned} p(x^{k+1}) &= \sum_{i=1}^{N_e} w_i^k \delta\Big(x^{k+1} - [f(x_i^k) + \beta_i^k]\Big) \\ &= \sum_{i=1}^{N_e} w_i^k \delta(x^{k+1} - x_i^{k+1}), \end{aligned} \tag{9}$$

i.e. $w_i^{k+1} = w_i^k$.

Now suppose we some observations of the system, y, at timestep n. What we desire is a representation of the posterior pdf at timestep n, $p(x^n \mid y)$. To do this we can use the weighted delta function representation of the prior in combination with Bayes' theorem (1):

$$p(x^n \mid y) = \sum_{i=1}^{N_e} \frac{w_i^n p(y \mid x_i^n)}{p(y)} \delta(x^n - x_i^n). \tag{10}$$

Hence the weights in the posterior pdf are the normalised product of the prior weights and the pointwise evaluation of the likelihood. For any subsequent timesteps, the posterior is used as the prior in a recursive manner.

Filter degeneracy, or weight collapse, is the case scenario in which $w_j^k \approx 1$ for some $j \in 1, \ldots, N_e$. Hence $w_i^k \approx 0\, \forall i \neq j$. In this case the first order moment of the posterior pdf, $\bar{x}^k$, will be simply x_j^k. All higher order moments will be computed to be approximately 0.

Snyder et al. (2008) showed that, in the case of using a naive particle filter such as the SIR filter (Gordon et al., 1993), to avoid filter degeneracy the number of ensemble members must be chosen such that $N_e \propto \exp(N_\tau^2)$ where N_τ is a measure of the dimension of the system. Ades and van Leeuwen (2013) showed that this dimension of the system is actually the number of independent observations.

Simply increasing the number of ensemble members is, for most geophysical applications, infeasible. N_e will be determined by the size of the supercomputer available. For operational NWP methods N_e may typically be around 50. For instance, simply for forecasting, the Canadian NWP ensemble forecast uses $N_e = 20$, ECMWF has $N_e = 51$ and the UK Met Office has $N_e = 46$.

Therefore it is clear that for a particle filter to represent the posterior pdf successfully the case that $w_j^k \approx 1$ for some $j \in 1, \ldots, N_e$ should be avoided. The equivalent weights particle filter (van Leeuwen, 2010) that we shall discuss in Section 2. is designed specifically so that $w_i^k \approx 1/N_e$ for all $i \in 1, \ldots, N_e$. It does this in a two stage process. Firstly the particles are nudged towards the observations. Secondly an "equivalent weights step" is made to avoid filter degeneracy.

1.3. Ensemble Kalman filters

The Ensemble Kalman filter (EnKF) is a method of data assimilation that attempts to solve Bayes' theorem when assuming that the posterior PDF is Gaussian (see for example (Evensen, 1994; Burgers et al., 1998; Evensen, 2007)). In that case, the posterior can be characterised by its first two moments: the mean and covariance. The prior pdf, or more precisely the covariance of the prior, is represented by an ensemble of model states. Instead of propagating the full covariance matrix of the prior by a numerical model (as in the Kalman Filter (Kalman, 1960)), only the ensemble members are propagated by the model.

If the dimension of the model state, N_x, is much greater than the number of ensemble members used, N_e, then the EnKF is much more computationally efficient than the Kalman filter.

Defining X_k to be the scaled matrix of perturbations of each ensemble member from the ensemble mean at time k, then the update equation of the EnKF can be written as

$$x_k^a = x_k^f + X_k X_k^T H^T (H X_k X_k^T H^T + R)^{-1} (y - H x_k^f). \tag{11}$$

Here, x_k^f refers to the $f$orecast of the ensemble member at time

k and x_k^a the resulting *a*nalysis ensemble member at time k which has been influenced by the observations. H is the observation operator which maps the model state into observation space and R is the observation error covariance matrix.

There are many different flavours of Ensemble Kalman filter, each of which is a different way to numerically compute the update equation. For a discussion on the different kinds see, for example, Tippett et al. (2003); Lei et al. (2010). In this paper we shall consider implementing the EnKF by means of the Local Ensemble Transform Kalman filter and shall discuss this in detail in Section 3.

1.4. Motivation for this investigation

We have seen that if we are trying to use a particle filter to recover the posterior pdf via a numerical implementation of Bayes' theorem then it makes sense to ensure the weights of each particle are approximately equal. Or at least, it pays to ensure that each particle has non-negligible weight, specifically when higher order moments of the posterior pdf are required.

Until now there has been no systematic comparison of the equivalent weights particle filter and an ensemble Kalman filter using a nontrivial model of fluid dynamics. This is a necessary study to see if anything is gained by not making the assumptions of Gaussianity that are made by the EnKF method. Previous investigations of the equivalent weights particle filter have focused on tuning the free parameters in the system to give appropriate rank histograms. In this study we shall investigate the method's ability to appropriately constrain the system in idealised twin experiments.

To this end the system we shall consider are the equations of fluid dynamics under the barotropic vorticity assumptions. This is perhaps the model most well studied for the equivalent weights particle filter. As a system of one prognostic variable on a 2-dimensional grid it is easily understood and reasonably cheap to experiment with. We also know the parameter regimes in which the equivalent weights particle filter will perform well.

The remainder of this paper is organised as follows. In Section 2. we discuss the use of proposal densities within particle filters before introducing the the equivalent weights particle filter. In Section 3. we discuss the Local Ensemble Transform Kalman filter. In Section 4. we discuss the barotropic vorticity model which we consider. In Section 5. we define the experimental setup which we use, and performance measures. In Section 6. we show the numerical results which are discussed in detail in Section 7. Finally in Section 8. we finish with some conclusions and discuss the implications for full-scale NWP.

2. Particle filters using proposal densities

In this section we briefly summarise the use of a proposal density within a particle filter, before going on to discuss the specific choices of these made in the equivalent weights particle filter.

2.1. Proposal densities

A key observation which has advanced the field of particle filters is the freedom to rewrite the transition density as

$$p(x^{k+1} \mid x^k) = \frac{p(x^{k+1} \mid x^k) q(x^{k+1} \mid x^k, y)}{q(x^{k+1} \mid x^k, y)} \tag{12}$$

which holds so long as the support of $q(x^{k+1} \mid x^k, y)$ is larger than that of $p(x^{k+1} \mid x^k)$. Now we are also free to change the dynamics of the system such that

$$x^{k+1} = f(x^k) + g(x^k, y) + \beta^k \tag{13}$$

as in van Leeuwen (2010). As in Section 1.2. we assume that, without loss of generality, we have a delta function representation for the prior at timestep k given by (2). Then, in a manner similar to the marginal particle filter (Klaas et al., 2005),

$$p(x^{k+1}) = \int \frac{p(x^{k+1} \mid x^k) q(x^{k+1} \mid x^k, y)}{q(x^{k+1} \mid x^k, y)} p(x^k) \, \mathrm{d}x^k \tag{14}$$

$$= \int \frac{p(x^{k+1} \mid x^k) q(x^{k+1} \mid x^k, y)}{q(x^{k+1} \mid x^k, y)} \sum_{i=1}^{N_e} w_i^k \delta(x^k - x_i^k) \, \mathrm{d}x^k \tag{15}$$

$$= \sum_{i=1}^{N_e} w_i^k \frac{p(x^{k+1} \mid x_i^k) q(x^{k+1} \mid x_i^k, y)}{q(x^{k+1} \mid x_i^k, y)}. \tag{16}$$

We can write the transition density $p(x^{k+1} \mid x_i^k)$ and proposal density $q(x^{k+1} \mid x_i^k, y)$ in terms of β^k:

$$p(x^{k+1}) = \sum_{i=1}^{N_e} w_i^k \int \frac{p(\beta^k) q(\beta^k)}{q(\beta^k)} \delta\Big(x^{k+1} - [f(x_i^k) + g(x_i^k) + \beta^k]\Big) \, \mathrm{d}\beta^k. \tag{17}$$

Now, similarly to before, drawing a single sample β_i^k for each ensemble member, but now from the distribution $q(\beta^k)$ gives

$$p(x^{k+1}) = \sum_{i=1}^{N_e} w_i^k \frac{p(\beta_i^k)}{q(\beta_i^k)} \delta\Big(x^{k+1} - [f(x_i^k) + g(x_i^k) + \beta_i^k]\Big) \tag{18}$$

$$= \sum_{i=1}^{N_e} w_i^k \frac{p(\beta_i^k)}{q(\beta_i^k)} \delta(x^{k+1} - x_i^{k+1}) \tag{19}$$

$$= \sum_{i=1}^{N_e} w_i^k \frac{p(x_i^{k+1} \mid x_i^k)}{q(x_i^{k+1} \mid x_i^k, y)} \delta(x^{k+1} - x_i^{k+1}). \tag{20}$$

i.e.

$$p(x^{k+1}) = \sum_{i=1}^{N_e} w_i^{k+1} \delta(x^{k+1} - x_i^{k+1}) \tag{21}$$

where

$$w_i^{k+1} = w_i^k \frac{p(x_i^{k+1} \mid x_i^k)}{q(x_i^{k+1} \mid x_i^k, y)}. \tag{22}$$

To find the delta function representation of the posterior, it is case of combining this derivation with Bayes' theorem to arrive at the same equation as in (10).

The use of proposal densities are the basis of particle filters such as the Implicit Particle Filter (Chorin and Tu, 2009) and the equivalent weights particle filter, and more recently the Implicit Equal Weights Filter (Zhu et al., 2016). The goal is to choose the proposal density in such a way that the weights w_i^k do not degenerate.

2.2. *The equivalent weights particle filter*

The equivalent weights particle filter (EWPF) is a fully nonlinear DA method that is nondegenerate by construction. For a comprehensive overview of the equivalent weights particle filter see van Leeuwen (2010) and Ades and van Leeuwen (2013).

A key feature of the EWPF is that it chooses the proposal density $q(x^{k+1} \mid x^k, y)$ equal to $p(\beta^k)$ but with new mean $g(x^k, y)$. It proceeds in a two-stage process with one form of $g(x^k, y)$ for the timesteps that have no observations and a different form of $g(x^k, y)$ when there are observations to be assimilated.

For each model timestep $k+1$ before an observation time n, the model state of each ensemble member, x_i^k, is updated via the equation

$$g(x_i^k, y) = A(y^n - H(x_i^k)) \tag{23}$$

where y^n is the next observation in time, H is the observation operator that maps the model state onto observation space and A is a relaxation term. In this work we consider

$$A = \sigma(k) Q H^T R^{-1} \tag{24}$$

where the matrices Q and R correspond to the model evolution error covariance and observation error covariance matrices respectively. $\sigma(k)$ is a function of the time between observations; in this paper $\sigma(k)$ increases linearly from 0 to a maximum (σ) at observation time. Equations (23) and (24) together make up what we will refer to as the nudging stage of the EWPF. This process is iterated until $k+1 = n-1$.

In this work we consider only unbiased Gaussian model error (i.e. $\beta_i^k \sim \mathcal{N}(0, Q)$). To obtain a formula for the un-normalised weights at timestep $k+1$, we can use this Gaussian form in (22). Taking logarithms leads to a formula for the weights of the particles (van Leeuwen, 2010; Ades and van Leeuwen, 2015) as

$$\begin{aligned} -\log(w_i^{k+1}) = &-\log(w_i^k) \\ &+ \frac{1}{2}(x_i^{k+1} - f(x_i^k))^T Q^{-1} (x_i^{k+1} - f(x_i^k)) \\ &- \frac{1}{2}(\beta_i^k)^T Q^{-1} (\beta_i^k). \end{aligned} \tag{25}$$

The second stage of the equivalent weights filter involves updating each ensemble member at the observation time n using the term

$$g(x_i^{n-1}, y) = \alpha_i Q H^T (H Q H^T + R)^{-1} (y^n - H(f(x_i^{n-1}))) \tag{26}$$

where α_i are scalars computed so as to make the weights of the particles equal. This is done for a given proportion ($0 < \kappa \leqslant 1$) of the ensemble which can make the desired weight. The remaining ensemble members are resampled using stochastic universal sampling (Baker, 1987; van Leeuwen, 2010).

It is important to realise that the covariance of the prior ensemble is never explicitly computed in the EWPF but implicitly, via the EWPF approximation to Bayes' theorem: increasing the spread in the prior will increase the spread in the posterior. Instead, the covariance of the error in the model evolution Q is crucial.

3. LETKF

The Local Ensemble Transform Kalman Filter (LETKF) is an implementation of the Ensemble Kalman filter which computes in observation space (Bishop et al., 2001; Wang et al., 2004; Hunt et al., 2007). As with all ensemble Kalman filters, the pdfs are assumed Gaussian. Formally, the LETKF update equation for ensemble member i at the observation timestep n can be written as

$$x_i^n = \overline{x}_f^n + X_f^n W_i^n \tag{27}$$

where $\overline{x}^f$ is the mean of the forecast ensemble, X^f the ensemble of forecast perturbations, and W_i^a is the column of a weighting matrix corresponding to ensemble member i. Full details of this is given in Hunt et al. (2007). This can be extended through time (Posselt and Bishop, 2012) such that for $k < n$, we get the Local Ensemble Kalman Smoother (LETKS) update equation

$$x_i^k = \overline{x}_f^k + X_f^k W_i^n. \tag{28}$$

As typically the number of ensemble members will be much fewer than the dimension of the model state, spurious correlations will occur within the ensemble. These spurious correlations lead to information from an observation inappropriately affecting the analysis at points far away from the observation. To counteract this, the LETKF effectively considers each point in the state vector separately and weights the observation error covariance by a factor depending on the distance of the observation from the point in the state vector.

For each point in the state vector, the inverse of the observation error covariance matrix, R^{-1} (also known as the precision matrix), is weighted by a function w so that

$$\hat{R}_{ij}^{-1} = R_{ij}^{-1} w(d(i))^{-1} w(d(j))^{-1}.$$

The weighting of the observation error covariance matrix R is given by the function

$$w(d)^{-1} = \begin{cases} \exp(-\frac{d^2}{4\ell^2}), & \text{if } \frac{d}{\ell} < 4 \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

where d is the distance between the point in the state vector and the observation and ℓ is a predefined localisation length-scale.

In the case of a diagonal R matrix, then

$$\hat{R}_{jj}^{-1} = R_{jj}^{-1} w(d(j))^{-2}.$$

The weighting $w(d)$ is a smoothly decaying function which cuts off when $\frac{d}{\ell} = 4$, i.e. $w(d)^{-2} = e^{-8} \approx 0.0003$. This means that the computations are speeded up by ignoring all the observations which have a precision less that 0.0003 of what they were originally.

Inflation is typically also required for the LETKF in large systems (e.g. Anderson and Anderson, 1999). That is, the ensemble perturbation matrices are multiplied by a factor of $(1+\rho)$ in order to increase the spread in the ensemble that is too small because of undersampling. i.e. $X^f \to (1+\rho)X^f$ in (27).

4. Barotropic vorticity model

In this section we consider the model which we investigate. We start with the Navier-Stokes equations and assume incompressible flow, no viscosity, no vertical flow and that flow is *barotropic* (i.e. $\rho = \rho(p)$). We define vorticity q to be the curl of the velocity field. This results in the following PDE in q (see for example Krishnamurti et al. (2006)), known as the barotropic vorticity (BV) model,

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} + v\frac{\partial q}{\partial y} = 0$$

where u is the component of velocity in the x direction and v is the component of velocity in the y direction. The domain we consider is periodic in both x and y and so the computation of this can be made highly efficient by the use of the FFT. In order to solve this equation, it is sufficient to treat vorticity q as the only prognostic variable. The curl operator can be inverted in order to derive the velocity field $\mathbf{u}$ from the vorticity. We use a 512×512 grid, making $N_x = 2^{18}$, a $262,144$ dimensional problem. Timestepping is achieved by a leapfrog scheme with $\mathrm{d}t = 0.04$ (roughly equivalent to a 15 minute timestep of a 22km resolution atmospheric model). The decorrelation timescale of this system is approximately 42 timesteps, or 1.68 time units.

There are a number of good reasons for investigating this model. For example, it exhibits strong nonlinear behaviour, develops cyclonic activity and generates fronts. All of which are typical of the highly chaotic regimes occurring in many meteorological examples. Turbulence in the model is prototypical: energy is transferred downscale due to the presence of nonlinear advection. See Figure 2a for a plot of a typical vorticity field from the model. Note that it was the barotropic vorticity model that was used for some of the earliest numerical weather predictions (Charney et al., 1950).

Note that this model has no balances that can be destroyed by data assimilation, something which should be considered in other studies of this kind. A further advantage for this first study is that we do not have to worry about bounded variables when applying the LETKF.

Also for this model we know the parameter regimes and model error covariance structure for which the EWPF performs well. Ades and van Leeuwen (2015) first applied the EWPF to the BV model, albeit at a lower resolution, and in this paper we employ similar parameters in the EWPF such as the nudging strength $\sigma(k)$ and use the same model error covariance matrix Q. The Ades and van Leeuwen (2015) study concentrated on using rank histograms as the performance diagnostic of the EWPF whereas in this paper we consider performance in terms of root mean squared errors.

5. Experimental setup

In this section we discuss the two experiments we shall run. All of the experiments were carried out using the EMPIRE data assimilation codes (Browne and Wilson, 2015) on ARCHER, the UK national supercomputer.

5.1. *Model error covariance matrix*

For ensemble methods in the NWP setting, obtaining spread in the ensemble is a key feature in the performance of both the analysis and the forecast. In NWP applications this is typically achieved by employing a stochastic physics approach (e.g. Baker et al., 2014) or using stochastic kinetic energy backscattering (e.g. Tennant et al., 2011) to add randomness at a scale which allows the model to remain stable. For the EWPF (or indeed any particle filter that uses a proposal density), we must specify (possibly implicitly) the model error covariance matrix. Understanding and specifying the covariances of model error in a practical model is a challenge to which much more research must be dedicated.

The model error covariance matrix used in this article is the same as that used in Ades and van Leeuwen (2015). That is, Q is a Second Order Autoregressive matrix based on the distance between two grid points, scaled so that the model error has a reasonable magnitude in comparison to the deterministic model step.

5.2. *Initial ensemble*

The initial ensemble is created by perturbing around a reference state. Thus, for each ensemble member x_i and the true state x_t,

$$\{x_i\}, x_t \sim \mathcal{N}(x_r, B) \qquad \forall i \in \{1, \dots, N_e\}. \qquad (30)$$

The background error covariance matrix B is chosen proportional to Q such that $B = 20^2 Q$. The reference state x_r is a random state which is different for each experiment.

5.3. *Truth run for twin experiments*

The instance of the model that is considered the truth is propagated forward using a stochastic version of the model where

$$x_t^{k+1} = f(x_t^k) + \beta_t \qquad \text{where} \qquad \beta_t^k \sim \mathcal{N}(0, Q).$$

5.4. Observing networks

We shall show results from experiments with 3 different observing networks that make direct observations of vorticity. The first is regular observations throughout the domain as considered by Ades and van Leeuwen (2015), the second a block of dense observations, and the third a set of strips which could be thought of as analogous to satellite tracks. The details of the observing networks are shown below and visualised in Figure 1.

ON1 Every other point in the x and y directions observed
ON2 Only those points such that $(x, y) \in [0, 0.5] \times [0, 0.5]$ are observed
ON3 Only those points such that $(x, y) \in [0, 1] \times ([0, 0.0675] \cup [0.25, 0.3175] \cup [0.5, 0.5675] \cup [0.75, 0.8175])$ are observed

In each case we have $N_y = N_x/4 = 65536$. The observation errors are uncorrelated, with a homogeneous variance such that $R = 0.05^2 I$. Observations occur every 50 model timesteps. These observations are quite accurate when you consider the vorticity typically lies in the interval $(-4, 4)$ (see Figure 2a).

5.5. Comparison runs

For comparison and analysis purposes we will run a number of different ensembles as well as the LETKF and the EWPF. We detail these subsequently.

5.5..1. Stochastic ensemble

Each ensemble member is propagated forward using a stochastic version of the model. That is,

$$x_i^{k+1} = f(x_i^k) + \beta_i \qquad \text{where} \qquad \beta_i^k \sim \mathcal{N}(0, Q).$$

5.5..2. Simple nudging

For each timestep, the nudging terms of the EWPF are used to propagate the model forward. That is, equations (13), (23) and (24) are used to update the model state. The weights of the particles are disregarded, and the ensemble is treated as if it was equally weighted.

5.5..3. Nudging with an LETKS relaxation

The model is propagated forward in time stochastically until the timestep before the observations. During this stage, no relaxation term is used (i.e. $g(x^k, y) = 0$). At the timestep before the observations, the relaxation term that is used comes from the LETKS. That is, term in (23) is the increment that would be applied by the LETKS. At the observation timestep, the ensemble is propagated using the stochastic model. The weights of the particles are disregarded, and the ensemble is treated as if it was equally weighted.

This can be written in equation form, so that at each iteration k before the observation time n, the update for each ensemble member i is given by

$$x_i^{k+1} = \begin{cases} f(x_i^k) + \beta_i^k & \text{for } k \in \{0, \ldots, n-3\} \cup \{n-1\} \\ f(x_i^k) + g_i + \beta_i^k & \text{for } k = n-2 \end{cases} \tag{31}$$

where g_i is the increment arising from the LETKS for ensemble member i.

5.5..4. The EWPF with an LETKS relaxation

Similarly to nudging with the LETKS relaxation, the model is propagated forward in time stochastically until the timestep before the observations. At the timestep before the observations, the relaxation that is used comes from the LETKS. At the observation timestep, the equivalent weights step (26) of the EWPF is used. The weights are calculated using (22) which in this case with Gaussian model error remains given explicitly by (25). We employ $\kappa = 0.75, 0.25$, and 0.5 for observations networks 1, 2, and 3 respectively. This is discussed in Section 7.3.

5.6. Assimilation experiments

Observations occur every 50 timesteps for the first 500 model timesteps. After that a forecast is made from each ensemble member for a further 500 timesteps.

For each observing network, we run 5 different experiments:

- The EWPF
- The LETKF
- Simple nudging
- Nudging with an LETKS relaxation
- The EWPF with an LETKS relaxation

Tables 1 and 2 list the parameter choices used for the different methods for the different observational networks. They were chosen by performing a parameter sweep across the various free parameters and selecting those that gave the lowest RMSEs (shown in Figure A1 in the appendix).

All of these experiments are repeated 11 times. In each of the 11 experiments, the initial reference state, x_r is different, as is the random seed used. For reference, we also run a stochastically forced ensemble from each of the different reference states. As no data is assimilated here, these runs are independent of the observing network.

We choose to run 48 ensemble members for each method. This is for 2 reasons: there are 24 processors per node on ARCHER so this is computationally convenient, and 48 is of the order of the number of ensemble members that operational NWP centres are currently using.

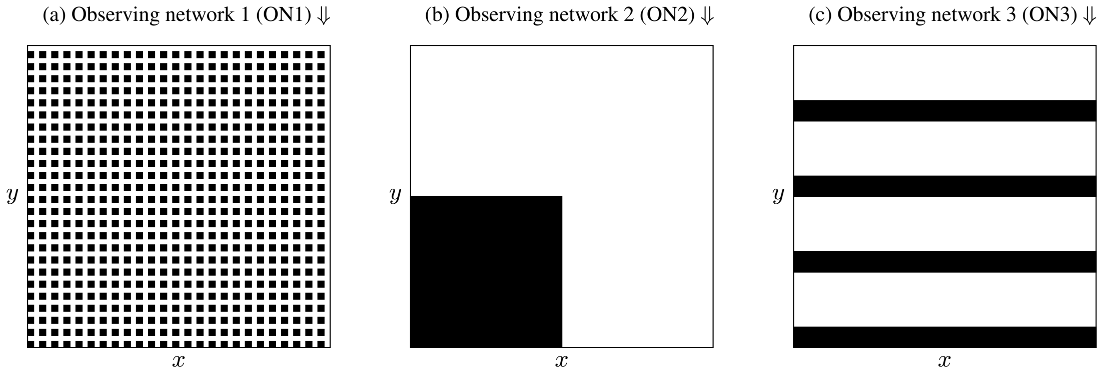

Fig. 1. Observing network diagrams

Table 1. Parameter values used in the LETKF

Observation network	on 1	on 2	on 3
Localisation length scale l	0.005	0.02	0.007
Inflation factor ρ	0.01	0.01	0.01

Table 2. Parameter values used in the EWPF

Observation network	on 1	on 2	on 3
Keep proportion κ	1.0	1.0	1.0
Nudging factor σ	0.7	0.5	0.7

6. Results

6.1. Root mean squared errors

Figures 3 to 5 show root mean squared errors (RMSE) for the different assimilation methods on the 3 separate observing networks. Formally, the RMSE we show is the square root of the spatial average of the square of the difference from the ensemble mean and the truth. Each line of the similar colour refers to a distinct experiment with a different stochastic forcing and initial reference state. Values are shown only for the initial ensemble, 10 analysis times (recall that each analysis time is separated by 50 model time steps) and 10 subsequent forecast times that are again separated by 50 model timesteps.

In brown, for reference, is plotted the RMSE from the stochastically forced ensemble. In black the total RMSE, blue the unobserved variables and red the observed variables.

The RMSE, as defined previously, is a measure of the similarity of the ensemble mean to the truth. If the posterior is a multimodal distribution then the ensemble mean may be far from a realistic, or accurate, state. EnKF methods, by their Gaussian assumptions that they make, naturally assume a unimodal posterior. Particle filters on the other hand do not make such an assumption. In this article we do not investigate the effect of using a different error measure.

Figure 3 is markedly different from Figures 4 and 5 - in this case the unobserved variables behave as if they are also observed. This is because each unobserved variable is either directly adjacent to 2 observed variables or diagonally adjacent to 4 observed variables. Contrast this with the observing networks 2 and 3 where an unobserved variable could be a maximum of 181 or 48 grid points, respectively, away from an observed variable.

6.2. Trajectories of individual gridpoints

In Figure 6 we show the evolution of the vorticity at a given gridpoint for a single experiment. Every model timestep is shown for each of the ensemble members for the different methods.

7. Discussion

It is clear from the results presented that the EWPF with simple nudging, as implemented by Ades and van Leeuwen (2015), is not competitive with the LETKF in terms of RMSEs. This is similar to the results noted in Browne and van Leeuwen (2015) in that the EWPF gives RMSEs higher than the error in the observations.

In this section we shall discuss different aspects of the results, in an attempt to give some intuition as to why they occur.

7.1. RMSEs from the EWPF are controlled by the nudging term

Consider the differences between RMSE plots for the simple nudging and the EWPF. They are qualitatively similar (Figures 3 - 5, (a) vs (c)). Further, when we use a different type of nudging (Figures 3 to 5, (d) vs (e)) the results are again similar.

This is due to the 2-stage nature of the EWPF. The first stage is a relaxation towards the observations (23), followed by a stage at observation time which ensures against filter degeneracy (26). In the second stage, we are not choosing the values of α_i to give a best estimate in some sense (compare with the Best Linear Unbiased Estimator, for example) but instead they are chosen so that the weights remain equal. Hence, most of the movement of the particles towards the observations happens in the first, relaxation, stage.

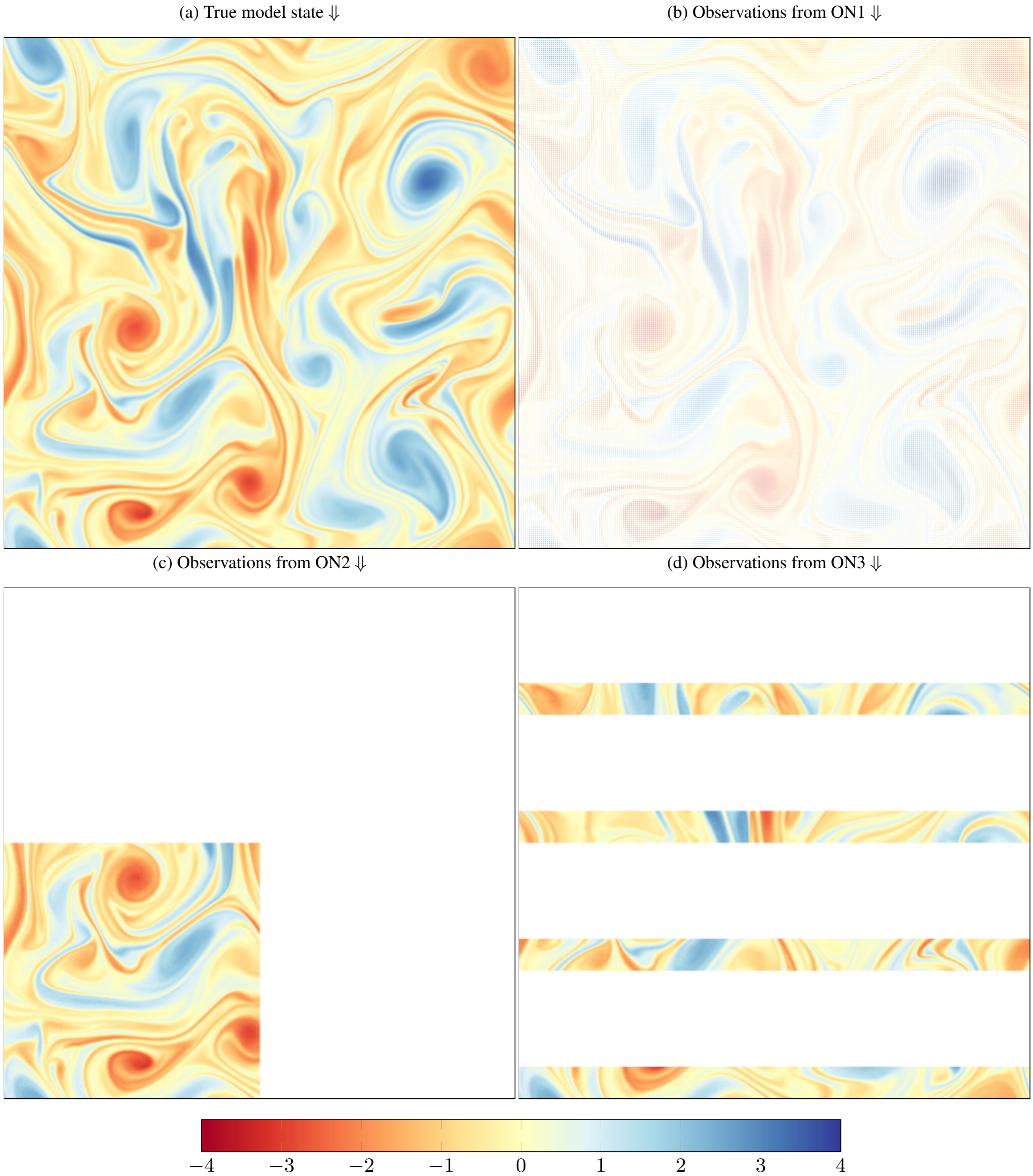

Fig. 2. Plots of vorticity for the true state and the resulting observations using the different networks at the 6th analysis time, for a particular random seed.

This is shown strongly in Figure 6; the simple nudging and the EWPF are qualitatively similar. Also in Figure 6 it can be seen that the LETKS nudging and the EWPF-LETKS also follow similar trajectories. This shows that the equivalent weights step of the EWPF is not moving the particles very far in state space in order to ensure the weights remain equal.

7.2. *Simple nudging is insufficient to get close to the observations*

Figures 3c to 5c show that, with simple nudging, the RMSEs are much larger that the observation error standard deviation. This is due to the choice of nudging equation used (24).

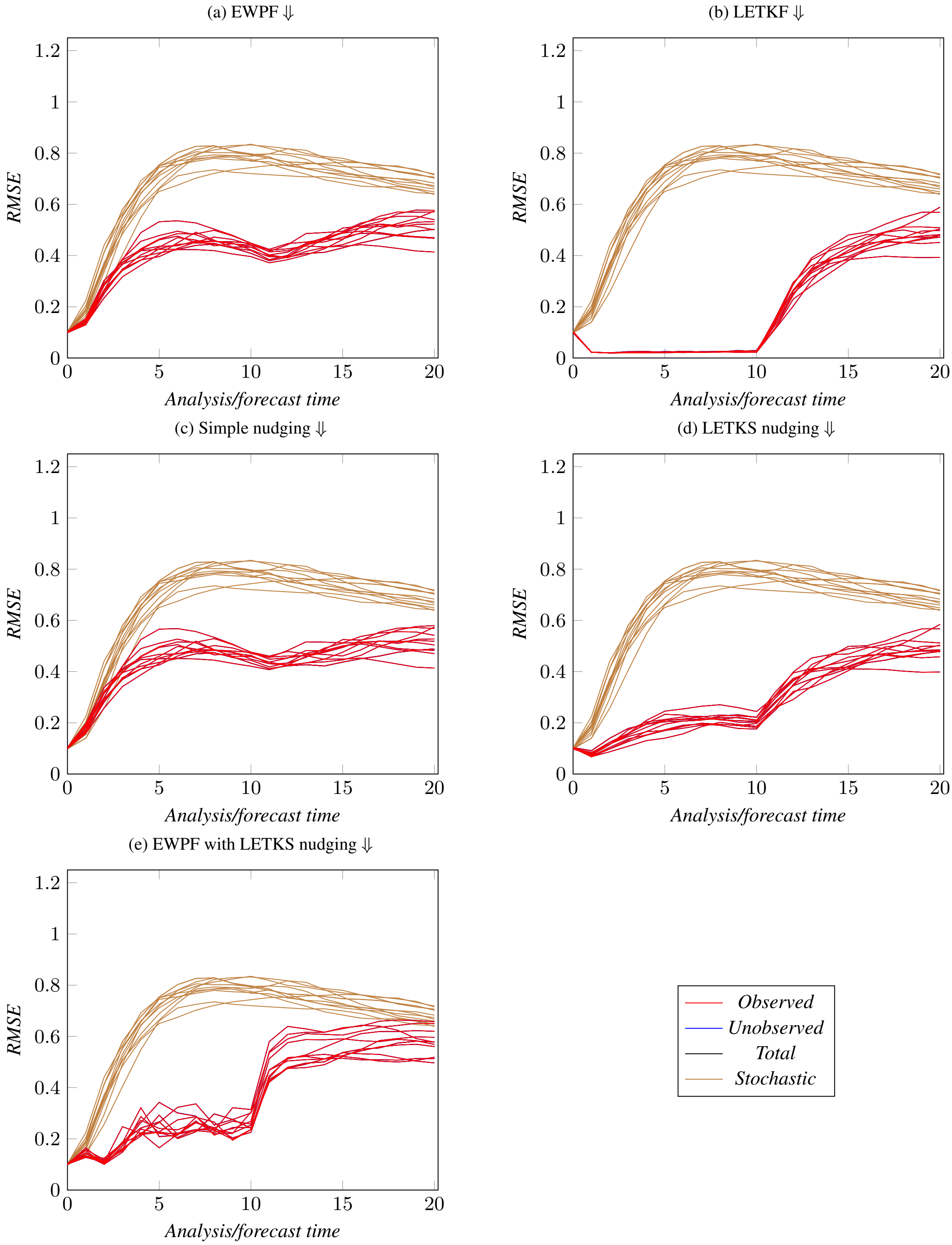

Fig. 3. Observing network 1, every other gridpoint. The Total and Unobserved RMSEs are almost exactly underneath the Observed RMSE plots. This is due to the widespread information from the observations effectively constraining the whole system.

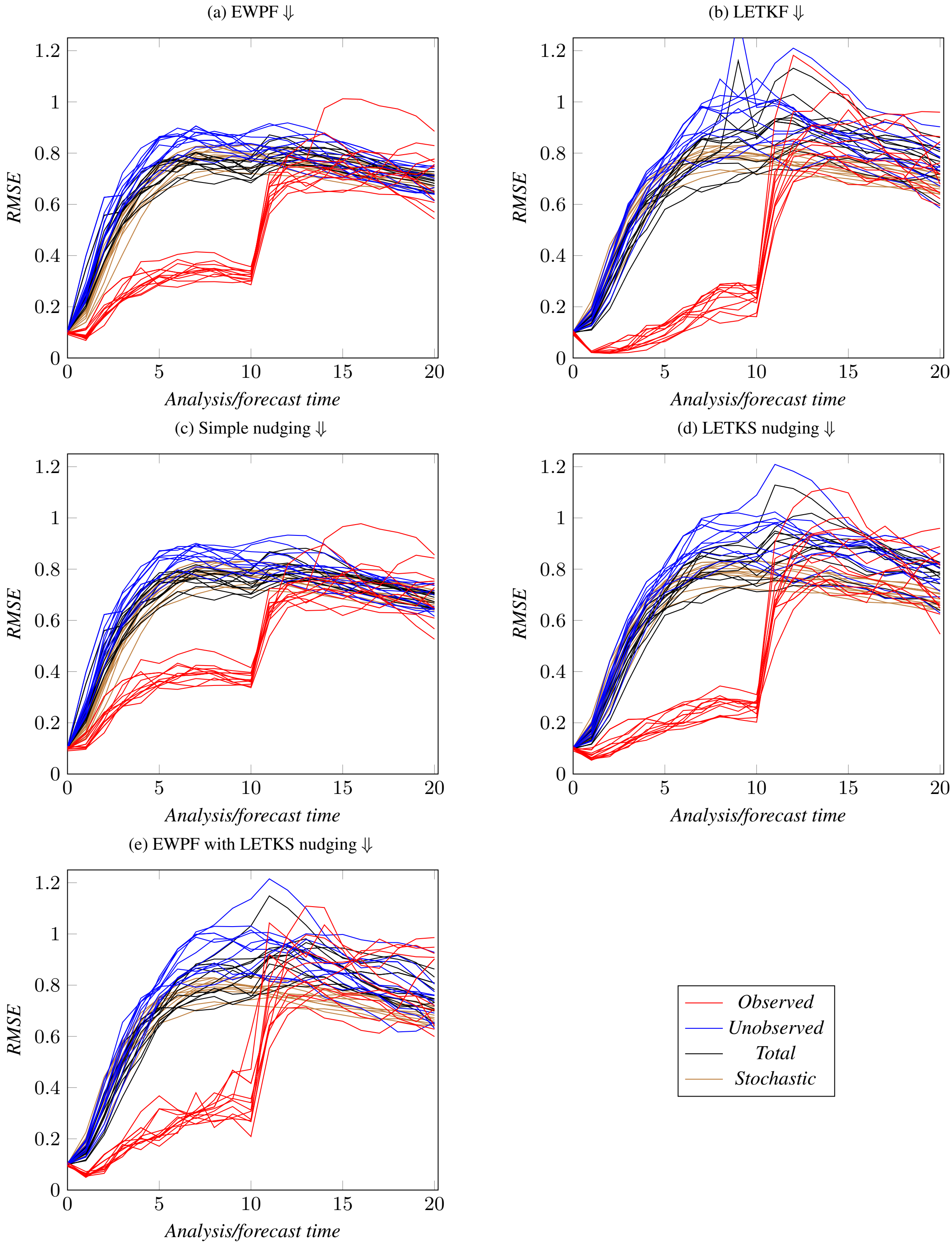

Fig. 4. Observing network 2, block of dense observations

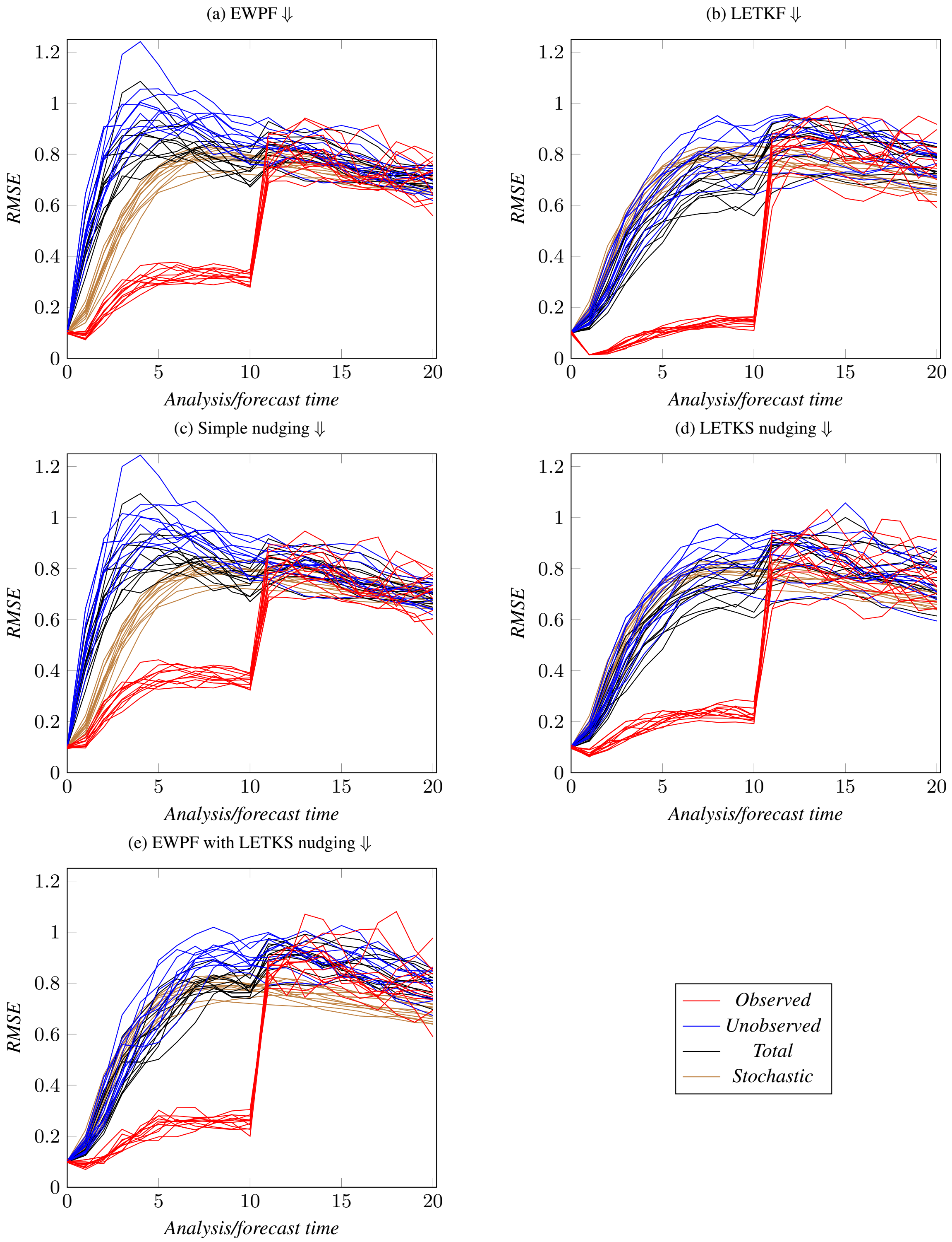

Fig. 5. Observing network 3, tracks of observations

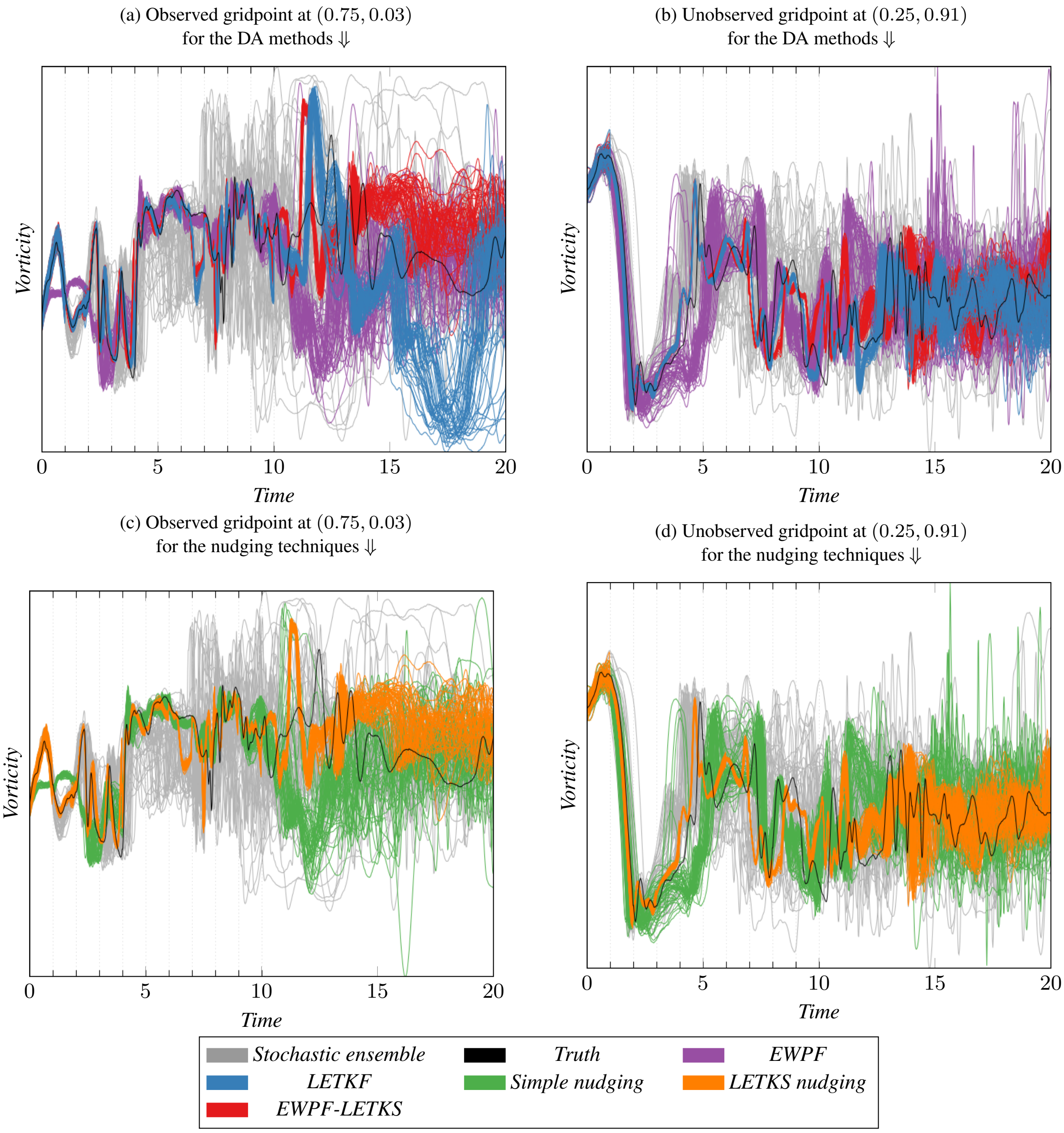

Fig. 6. Trajectories of 2 different points in the domain when using the different assimilation methods with observing network 3 for a single experiment

The goal of nudging is to bring the particles closer to the observations, or equivalently, to the area of high probability in the posterior distribution. In this section we shall discuss the properties that this nudging term should have.

Let the nudging term be denoted $A(x, y)$ and write it as a product of operators

$$A(x, y) = A_s \circ A_m \circ A_w \circ A_I$$

where A_I is the innovation, A_w is the innovation weighting, A_m a mapping from observation space to state space and A_s an operator to spread the information from observation space throughout state space.

The innovation should have the form

$$A_I = y - H(f(x))$$

where f takes the state at the current time and propagates it forward to the corresponding observation time. With this, the innovation is exactly the discrepancy in observation space that we wish to reduce, however it is valid only at the observation time.

Consider now the innovation weighting A_w. When the observations are perfect we wish to trust them completely and hence we should nudge precisely to the observations. When they are poor, we should distrust them and nudge much less strongly to the observations. Hence

$$R \to 0 \implies A_w \to I \qquad \& \qquad R \to \infty \implies A_w \to 0.$$

Hence with

$$A_w = (I + R)^{-1}$$

the appropriate limits are obtained.

$A_m = H^T$ is a way to map the scaled innovations into state space.

The term A_s should compute what increment at the current time would lead to such an increment at observation time. Hence $A_s = M^T$, the adjoint of the forward model.

Thus to nudge consistently,

$$A(x, y) = A_s \circ A_m \circ A_w \circ A_I = M^T H^T (I+R)^{-1}[y - H(f(x))] \tag{32}$$

Now let us compare this to the simple nudging term in (23), working through the terms from right to left.

$$A_I = y - H(f(x)) \neq y - H(x) \tag{33}$$

In the simple nudging term, the innovations used compare the observations with the model equivalent at the current time. This ignores the model's evolution in the intervening time, and thus the more the model evolves, the larger this discrepancy. This discrepancy occurs even with linear model evolution. In Figure 6 this can be seen by considering the evolution of the simple nudging ensemble between times 0 and 1. The model is forced to be close to the observation too early due to this time discrepancy in the innovation.

$$A_w = (I + R)^{-1} \neq R^{-1}$$

For the form of observation error covariance matrix R used in this study, this is not an issue. To see this, we have to consider $A_w = \sigma R$, and note that we have $R = \gamma I$. Then $I + R = I + \gamma I = (1 + \gamma)I$, and hence $I + R = \frac{(1+\gamma)}{\gamma} R$. Thus the coefficient $\frac{(1+\gamma)}{\gamma}$ can be subsumed into the nudging coefficient σ.

With simple nudging A_m is consistent.

Finally, the term $A_s = M^T \neq Q$. The model error covariance matrix is clearly not a good approximation to the adjoint of the model. Hence the information from the observations is not propagated backwards in time consistently.

All of these factors serve to make simple nudging ineffective at bringing the ensemble close to the observations.

7.3. LETKS as a relaxation in the EWPF

Given the theory described in Section 7.2., it is reasonable to believe that the Ensemble Kalman Smoother (EnKS) may provide better information with which to nudge.

As with the EnKF, there are many flavours of EnKS. Here we have used the LETKS simply because of its availability within EMPIRE.

Using the notation of the EnKF introduced in Section 1.3., we can write the EnKS analysis equation as

$$x_\ell^a = x_\ell^f + X_\ell^f X_k^{fT} H^T (H X_k^f X_k^{fT} H^T + R)^{-1} (y - H x_k^f). \tag{34}$$

Hence the nudging term that comes from the EnKS is

$$g(x^k, y) = X_\ell^f X_k^{fT} H^T (H X_k^f X_k^{fT} H^T + R)^{-1} (y - H x_k^f).$$

Comparing with (32), we can see that the innovations are correct. The observation error covariance matrix is regularised with $H X_k^f X_k^{fT} H^T$ instead of the identity, but the same limits are reached as $R \to 0$ and $R \to \infty$. The main difference is that now the information is brought backwards in time via the temporal cross-covariances of the state at the current time and the forecasted state at the observation time. Hence using this method there is no need for the model adjoint.

Comparing Figures 3 to 5, (c) vs (d) it can be seen that LETKS nudging provides a decrease in RMSE when compared to the simple nudging. Moreover, comparing the trajectories shown in Figures 6c and 6d it can be seen that the LETKS nudging follows the evolution of the truth much more closely than the simple nudging. This is especially noticeable at the timesteps between observations, likely due to the time discrepancy of the innovations that simple nudging makes (see equation (33)).

There are immediate extra computational expenses involved with using the LETKS as a nudging term. Firstly, the model has to be propagated forward to the observation time in order to find the appropriate innovations. Secondly, the LETKF has to be used to calculate the nudging terms, thus adding a large amount to the computational cost.

Moreover, consider the difference in the weight calculations caused by using the LETKS and not the simple nudging given in (23) and (24). Writing the update equation in the form

$$x_i^{k+1} = f(x_i^k) + g_i + \beta_i \tag{35}$$

where g_i is the nudging increment and β_i is a random term. The weight update has the form (van Leeuwen, 2010; Ades and van Leeuwen, 2015):

$$\begin{aligned} -\log(w_i^{k+1}) = &-\log(w_i^k) \\ &+ \tfrac{1}{2}(g_i + \beta_i)^T Q^{-1} (g_i + \beta_i) \\ &- \tfrac{1}{2}\beta_i^T Q^{-1} \beta_i. \end{aligned} \tag{36}$$

When $\beta_i \sim \mathcal{N}(0, Q)$, $\beta_i = Q^{\frac{1}{2}} \eta_i$ where $\eta_i \sim \mathcal{N}(0, I)$. Hence the final term

$$\beta_i^T Q^{-1} \beta_i = \eta_i^T Q^{\frac{T}{2}} Q^{-1} Q^{\frac{1}{2}} \eta_i = \eta_i^T \eta_i \tag{37}$$

can be calculated without a linear solve with Q. Similarly, if the nudging term g_i is premultiplied by Q (or $Q^{\frac{1}{2}}$) then Q^{-1} cancels in the calculation of the weights. This is the case for the simple nudging used as given in (23).

Hence, using the LETKS to compute a nudging term for use in a particle filter, we cannot avoid computing with Q^{-1} to find the appropriate weights for each ensemble member. This may prove to be prohibitive for large models, or must be a key consideration in the choice of Q matrix used. In the application to the BV model shown in this article, Q is computed in spectral space using the FFT, hence applying any power of Q to a vector is effectively the same computational cost.

Furthermore, in order to compute the LETKS nudging term, EnKF-like arguments are adopted. That is, the when computing the analysis update, the posterior pdf is assumed Gaussian. Linear model evolution is assumed so that the updates can be propagated backwards in time. Having made this Gaussian assumption at the timestep before the observations will limit the benefits of using the fully nonlinear particle filter which does not make any such assumptions on the distribution of the posterior. Indeed, considering the evolution of the EWPF with the LETKS nudging and comparing with that of the LETKF (Figures 6a and 6b), they are markedly similar. Hence the extra expense of the EWPF over the LETKF may not be justified.

The choice of κ when we use the LETKS as a relaxation within the EWPF is a complicated and not fully understood process. Figures B1 to B4 in the Appendix show the behaviour of the analysis as you vary κ for each different observation network. What is clear is that the optimal κ is problem dependent. Further, it can be seen that $\kappa = 1$ performs poorly in all cases. One conjecture for this is that using the LETKS as a relaxation gives a large change to some ensemble members. Making a large change to the position of any ensemble member must be paid for in the weights of that particle: its weight decreases. Keeping $\kappa = 1$ forces all ensemble members to degrade their positions in order to achieve a weight equal to that of the worst particle. This process could then move all the other ensemble members away from the truth – thus increasing the RMSE. Further investigations on this matter are warranted.

8. Conclusions

Both the Local Ensemble Transform Kalman Filter and the Equivalent Weights Particle filter were used in data assimilation experiments with the barotropic vorticity model. Typical values for the parameters in the methods were used for 3 different set of observations.

In all cases, the LETKF was found to give RMSEs that were substantially smaller than those achieved by the EWPF. Notably, the EWPF gives RMSEs much larger than that of the observation error standard deviations.

The efficacy of the EWPF to minimise the RMSE was shown to be controlled by the nudging stage of the method. Experiments with both simple nudging and using the LETKS as a relaxation showed that the resulting particle filter followed those trajectories closely. An analysis of the relaxation term used in the simple nudging procedure showed why such a method does not bring the ensemble mean close to the truth. This same analysis motivated the use of the LETKS relaxation and this was numerically shown to lead to improvements in RMSE.

The model investigated had a state dimension of $N_x = 262144$ and assimilated $N_y = 65536$ observations at each analysis. In such a high-dimensional system it is a challenge to ascertain if the posterior is non-Gaussian. Without such knowledge it appears that the LETKF is a better method of data assimilation in terms of efficiency and accuracy.

Finally, note that all these experiments were conducted with an ensemble size of $N_e = 48$. This ensemble size is representative of what can typically be run operationally. In the future, if much larger ensembles are affordable, then the results presented here may be different when the data assimilation methods are tuned to a significantly larger ensemble size.

9. Acknowledgements

The author would like to thank Chris Snyder for his insightful questioning into the effectiveness of the EWPF and Keith Haines for his questions regarding the practical implementation of the EWPF within a reanalysis system. Both lines of inquiry showed the need to perform the investigations in this paper.

The author would also like to acknowledge Javier Amezcua and Peter Jan van Leeuwen for their valuable discussions.

This work was supported by NERC grant NE/J005878/1. This work used the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk).

References

Ades, M. and van Leeuwen, P. (2013). An exploration of the equivalent weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 139(672):820–840.

Ades, M. and van Leeuwen, P. (2015). The equivalent-weights particle filter in a high dimensional system. *Quarterly Journal of the Royal Meteorological Society*, 141(687):484–503.

Anderson, J. L. and Anderson, S. L. (1999). A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts. *Monthly Weather Review*, 127(12):2741–2758.

Baker, J. E. (1987). Reducing Bias and Inefficiency in the Selection Algorithms. In *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, pages 14–21, Hillsdale, New Jersey, US. Lawrence Erlbaum Associates.

Baker, L. H., Rudd, a. C., Migliorini, S., and Bannister, R. N. (2014). Representation of model error in a convective-scale ensemble prediction system. *Nonlinear Processes in Geophysics*, 21(1):19–39.

Bayes and Price (1763). An Essay towards solving a Problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, F.R.S Commicated by Mr. Price, in a letter to John Canton, A.M.F.R.S. *Philosophical Transactions*, 53:370–418.

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(3):420–436.

Browne, P. and van Leeuwen, P. (2015). Twin experiments with the equivalent weights particle filter and HadCM3. *Quarterly Journal* *of the Royal Meteorological Society*, 141(693 October 2015 Part B):3399–3414.

Browne, P. and Wilson, S. (2015). A simple method for integrating a complex model into an ensemble data assimilation system using MPI. *Environmental Modelling & Software*, 68:122–128.

Burgers, G., Jan van Leeuwen, P., and Evensen, G. (1998). Analysis Scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, 126(6):1719–1724.

Charney, J. G., Fjörtoft, R., and Neumann, J. V. (1950). Numerical Integration of the Barotropic Vorticity Equation. *Tellus A*, 2(4):238–254.

Chorin, A. J. and Tu, X. (2009). Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences of the United States of America*, 106(41):17249–17254.

Dashti, M., Law, K. J. H., Stuart, a. M., and Voss, J. (2013). MAP estimators and their consistency in Bayesian nonparametric inverse problems. *Inverse Problems*, 29(9):095017.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans (1978–2012)*, 99(C5):10143–10162.

Evensen, G. (2007). *Data assimilation*. Springer.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113.

Hunt, B. R., Kostelich, E. J., and Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, 230(1-2):112–126.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems.

Klaas, M., de Freitas, N., and Doucet, A. (2005). Toward Practical N^2 Monte Carlo: The Marginal Particle Filter. *Proceedings of the Twenty-First Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 308–315.

Krishnamurti, T., Bedi, H., Hardiker, V., and Watson-Ramaswamy, L. (2006). *An Introduction to Global Spectral Modeling*. Atmospheric and Oceanographic Sciences Library. Springer New York.

Le Dimet, F.-X. and Talagrand, O. (1986). Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, 38A(2):97–110.

Lei, J., Bickel, P., and Snyder, C. (2010). Comparison of Ensemble Kalman Filters under Non-Gaussianity. *Monthly Weather Review*, 138(4):1293–1306.

Posselt, D. J. and Bishop, C. H. (2012). Nonlinear Parameter Estimation: Comparison of an Ensemble Kalman Smoother with a Markov Chain Monte Carlo Algorithm. *Monthly Weather Review*, 140(6):1957–1974.

Smith, A., Doucet, A., de Freitas, N., and Gordon, N. (2013). *Sequential Monte Carlo methods in practice*. Springer Science & Business Media.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008). Obstacles to High-Dimensional Particle Filtering. *Monthly Weather Review*, 136(12):4629–4640.

Tennant, W. J., Shutts, G. J., Arribas, A., and Thompson, S. a. (2011). Using a Stochastic Kinetic Energy Backscatter Scheme to Improve MOGREPS Probabilistic Forecast Skill. *Monthly Weather Review*, 139(Mittermaier 2007):1190–1206.

Tippett, M. K., Anderson, J. L., Bishop, C. H., Hamill, T. M., and Whitaker, J. S. (2003). Ensemble Square Root Filters. *Monthly Weather Review*, 131(7):1485–1490.

van Leeuwen, P. (2010). Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136(653):1991–1999.

van Leeuwen, P. J. (2009). Particle Filtering in Geophysical Systems. *Monthly Weather Review*, 137(12):4089–4114.

Wang, X., Bishop, C. H., and Julier, S. J. (2004). Which Is Better , an Ensemble of Positive-Negative Pairs or a Centered Spherical Simplex Ensemble? *Monthly Weather Review*, 132:1590–1605.

Zhu, M., Leeuwen, P. J. V., and Amezcua, J. (2016). Implicit Equal-Weights Particle Filter. *Quarterly Journal of the Royal Meteorological Society*.

APPENDIX A: EWPF parameter sensitivity

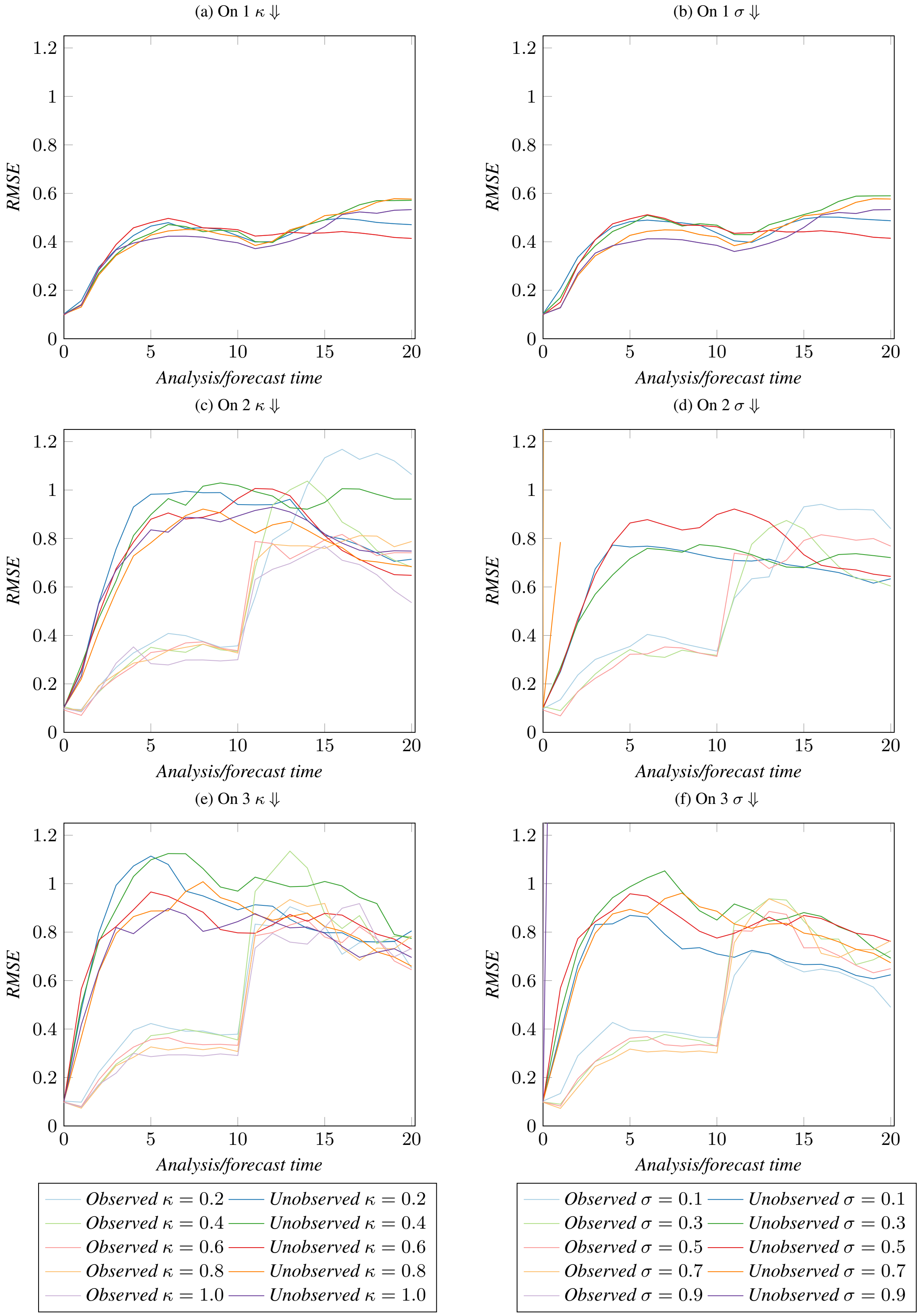

Fig. A1. Performance of the EWPF under different parameters

APPENDIX B: EWPF with LETKS relaxation

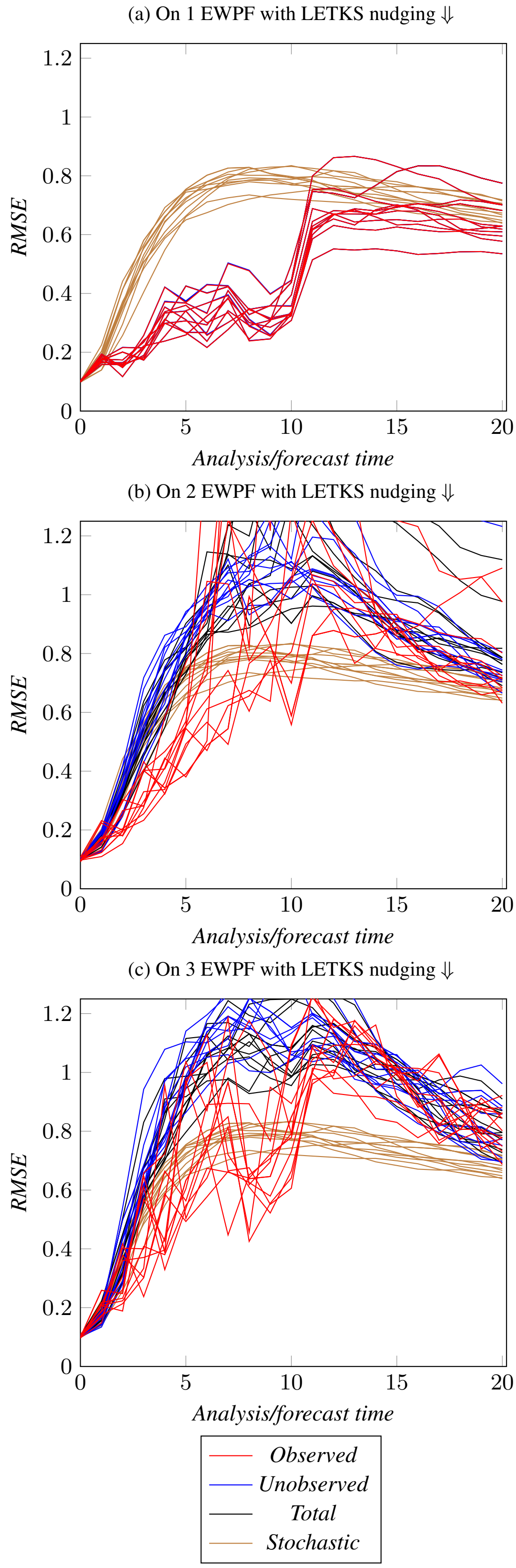

Fig. B1. Performance of the EWPF with the LETKS relaxation when $\kappa = 1.0$

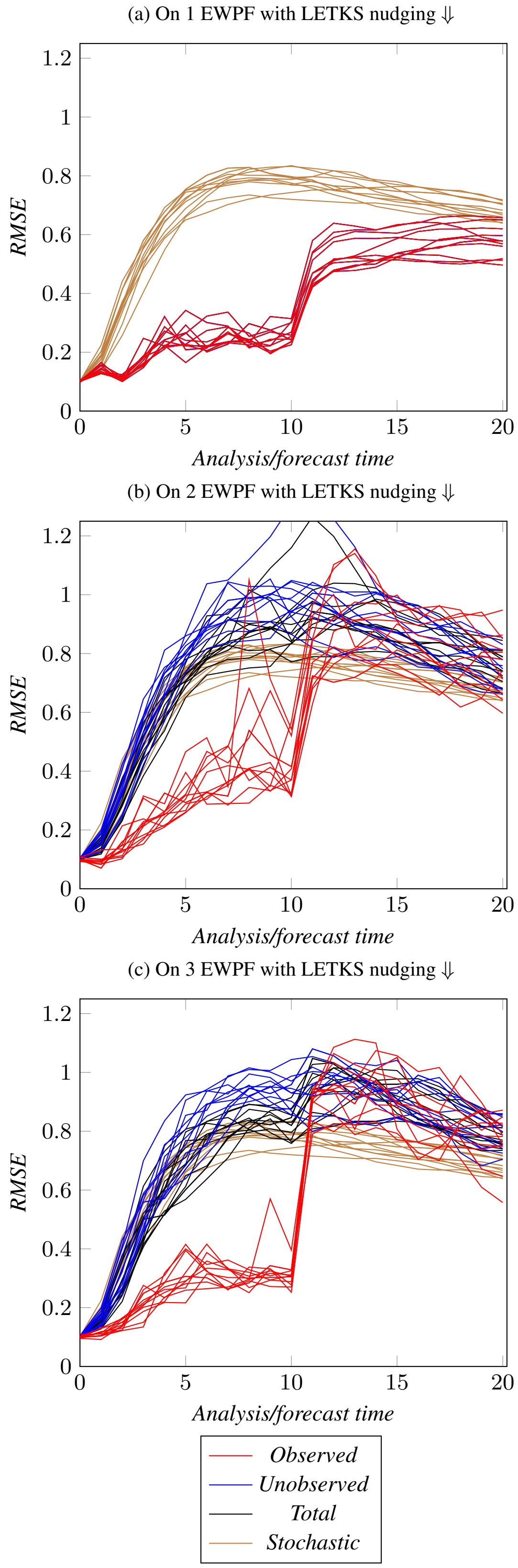

Fig. B2. Performance of the EWPF with the LETKS relaxation when $\kappa = 0.75$

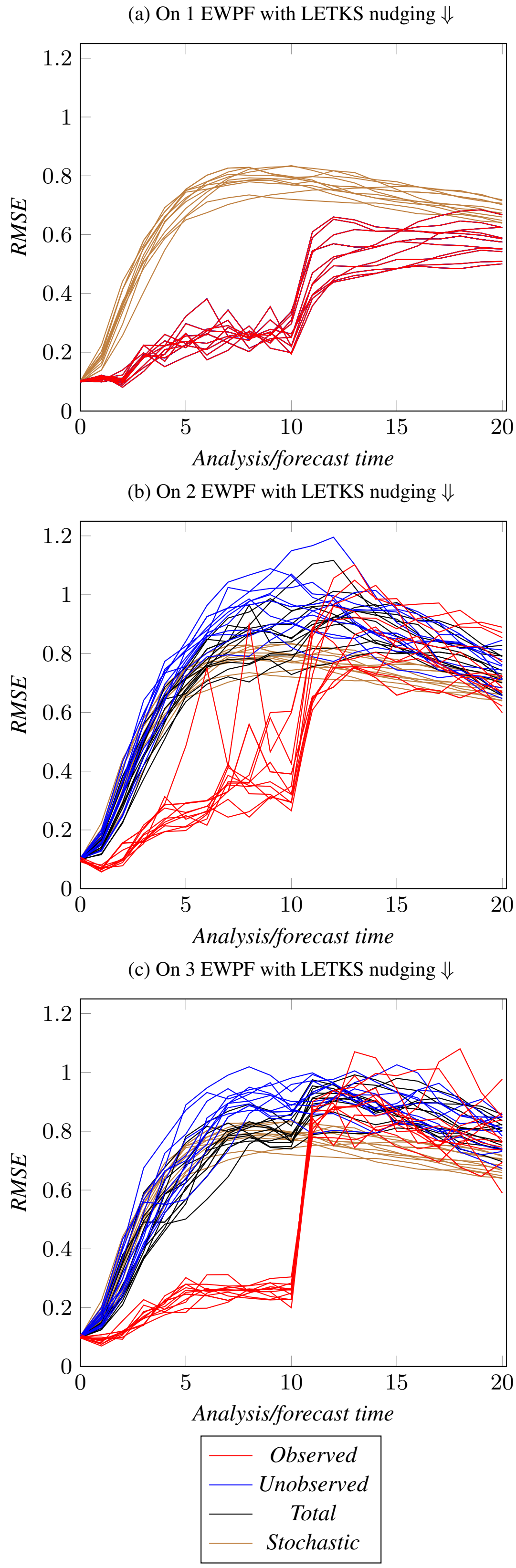

Fig. B3. Performance of the EWPF with the LETKS relaxation when $\kappa = 0.50$

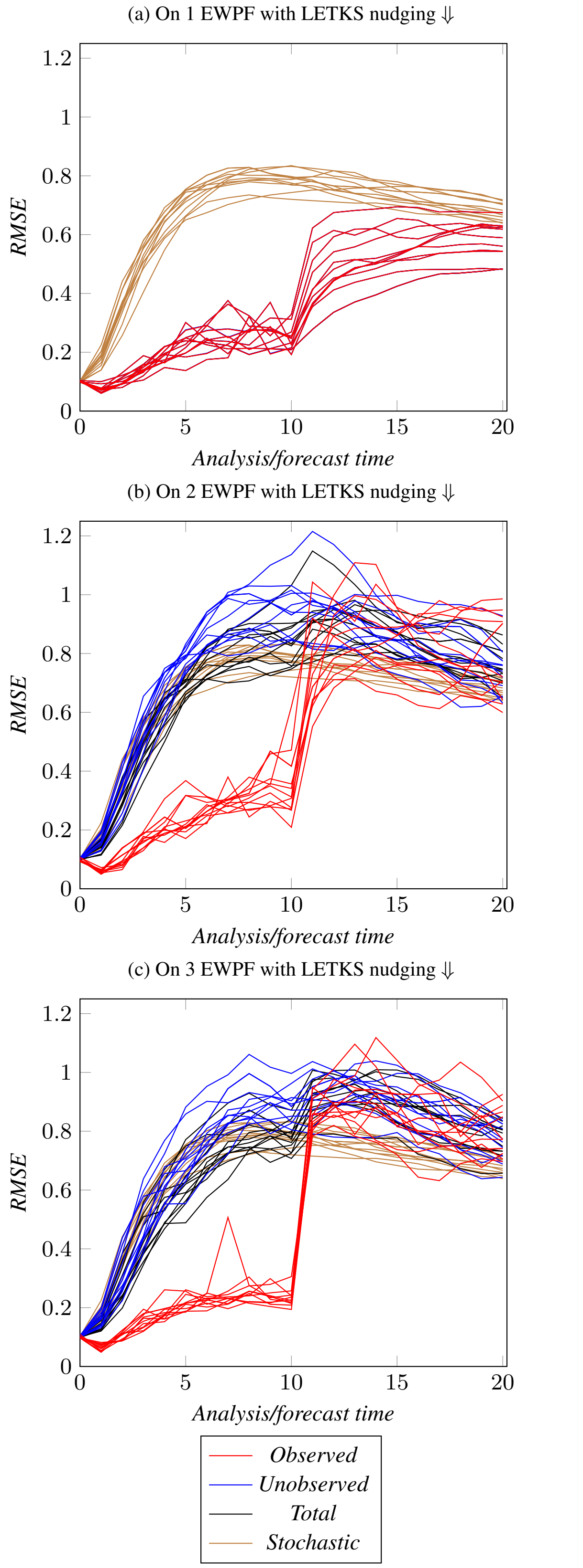

Fig. B4. Performance of the EWPF with the LETKS relaxation when $\kappa = 0.25$