Parallel Direct Solution of the Covariance-Localized Ensemble Square-Root Kalman Filter Equations with Matrix Functions

Jeffrey L. Steward*

University of California Los Angeles, Los Angeles, California, United States

Jose E. Roman

Universitat Poltècnica de València, València, Spain

Alejandro Lamas Daviña

Universitat Poltècnica de València, València, Spain

Altuğ Aksoy

(1) Cooperative Institute for Marine and Atmospheric Studies, University of Miami, Miami, Florida, United States, (2) Hurricane Research Division, NOAA/AOML, Miami, Florida, United States

**Corresponding author address:* Joint Institute for Regional Earth Science and Systems Engineering, University of California, Los Angeles, Los Angeles, California, United States

E-mail: jsteward@jifresse.ucla.edu

ABSTRACT

Recently, the serial approach to solving the Square-Root Ensemble Kalman Filter (ESRF) equations in the presence of covariance localization was found to depend on the order of observations. As shown previously, correctly updating the localized posterior covariance in serial requires additional effort and computational expense. A recent work, Steward et al. (2017), details an all-at-once direct method to solve the ESRF equations in parallel. This method uses the eigenvectors and eigenvalues of the forward observation covariance matrix to solve the difficult portion of the ESRF equations. The remaining assimilation is easily parallelized, and the analysis does not depend on the order of observations. While this allows for long localization lengths that would render local analysis methods inefficient, in theory an eigenpair-based method scales as the cube number of observations, making it infeasible for large numbers of observations. In this work, we extend this method to use the theory of matrix functions to avoid eigenpair computations. The Arnoldi process is used to evaluate the covariance localized ESRF equations on the reduced-order Krylov subspace basis. This method is shown to converge quickly and apparently regains a linear scaling with the number of observations. The method scales similarly to the widely-used serial approach of Anderson and Collins (2007) in wall-time but not in memory usage. To improve the memory usage issue, this method potentially can be used without an explicit matrix. In addition, hybrid ensemble and climatological covariances can be incorporated.

1. Introduction

Data assimilation of increasingly plentiful satellite and radar observations requires efficient and accurate algorithms. A single overpass of a polar orbiting satellite over a regional numerical weather prediction (NWP) domain can produce tens of thousands of potentially usable observations, especially when all-sky observations are considered. The Japanese K computer assimilates radar observations every 30 seconds with a 100-m grid spacing (Miyoshi et al. 2016), and with the next generation GOES-16 (Schmit et al. 2016) and Himawari 8 (Bessho et al. 2016) geostationary observing platforms providing observations with approximately kilometer resolution approximately every 5 minutes, data assimilation algorithms need to handle increasingly large data volumes to keep pace. In this paper we describe a new, efficient, and parallel technique for solving the covariance-localized Square-Root Ensemble Kalman Filter equations that overcomes several issues in previously described implementations.

The Ensemble Kalman Filter, first introduced in Evensen (1994), is one of the most widely used methods for data assimilation. Using an ensemble with a relatively small number of members to estimate the flow-dependent background error covariance from the Kalman filter as originally formulated (Kalman 1960) made it feasible to run statistical data assimilation problems even on very large domains. However, two main issues became apparent in the implementation of the Ensemble Kalman Filter. The first is that using the same observations to update the mean and ensemble perturbations leads to a systematic underestimation of covariance. Secondly, the unlocalized estimated covariances contain sample error due to the low number of ensemble members used, leading to spurious relationships.

The issue of systematic covariance underestimation was first solved by perturbing observations with independently sampled noise for each ensemble member (Houtekamer and Mitchell 1998;

Burgers et al. 1998). While this solves the underestimation of covariance, adding additional noise increases sampling error, causing the filter to be suboptimal especially when the ensemble size is small (Whitaker and Hamill 2002). Subsequently, the Ensemble Square-Root Filter (ESRF) was introduced that corrects for the under-representation of error covariance by adding a square-root term to the Kalman update for the ensemble. Various flavors of ESRF have been developed (Bishop et al. 2001; Anderson 2001; Whitaker and Hamill 2002), which Tippett et al. (2003) showed are all equivalent in the sense they perform analysis in the same vector space and find the same covariance. These methods as originally formulated assume the rank of the covariance matrices is the number of ensemble members.

Independently from covariance underestimation, the issue of spurious correlations due to small ensemble size has been addressed in two main ways: covariance localization and local analysis. Sakov and Bertino (2011) demonstrated that these two approaches are approximately equal, and the choice of approach is therefore dependent upon other factors. Critically, the localization radius used in local methods will determine their efficiency, and large localization radii will require repetitive solution of large problems for each grid point. In this work we investigate covariance localization, which uses a Schur product (component-wise multiplication) to zero out correlations further than a specified distance (Gaspari and Cohn 1999; Houtekamer and Mitchell 2001; Hamill et al. 2001). This causes the rank of the forward-observation-covariance matrix used in the inverse of the Kalman gain to increase beyond the number of ensemble members. As shown in Steward et al. (2017), a relatively short localization radius will lead to a full-rank forward-observation-covariance matrix, while a long localization radius will lead to a rank deficient one.

The combination of these factors leads to several different possibilities for scalable parallel implementations of the Ensemble Kalman filter equations. Local methods with perturbed observations and covariance localization include Keppenne and Rienecker (2002); Houtekamer et al.

(2013); Bishop et al. (2015); Nino-Ruiz et al. (2015), while local analysis methods based on the ESRF equations include Ott et al. (2002); Anderson (2003); Zhang et al. (2005); Hunt et al. (2007); Wang et al. (2013); Nino-Ruiz et al. (2017). Note that the widely used Local Ensemble Kalman Transform Filter of Hunt et al. (2007, LETKF) applies a localization strategy based on the observation error covariance matrix $\mathbf{R}$ rather than on the sample covariance matrices estimated by the ensemble. The widely-used and highly efficient method of Anderson and Collins (2007) is a "global" analysis (i.e. non-local) parallel implementation based on the serial assimilation of the ESRF equations with covariance localization. This method also treats the observations as part of an augmented state in order to update the observations in parallel without requiring excessive communication. Houtekamer and Mitchell (2001) describes a global analysis method with perturbed observations and covariance localization.

Due to the difficulties in solving the global ESRF equations directly, in implementations such as Anderson (2001), Whitaker and Hamill (2002), Anderson and Collins (2007) and Aksoy (2013) a serial approach is utilized where a single observation is assimilated at a time. This approach is provably identical to the global analysis without covariance localization and linear observation operators. However, with covariance-based localization, the ordering of observations affects the analysis as shown in Nerger (2015) and Bishop et al. (2015) due to the nonlinear nature of covariance localization. In other words, in the presence of ensemble sample covariance localization, serially assimilating observation A before observation B may give different results than assimilating observation B before A. The magnitude of this issue has not yet been fully explored.

As shown in Bishop et al. (2015), the issue of observation-ordering dependent analysis in serial covariance localized methods stems from the inconsistent application of the high-rank localized covariance matrices. In particular, when covariance localization is used, the matrix to invert in the Kalman gain becomes full rank or nearly full-rank as shown in e.g. Steward et al. (2017).

Without covariance localization (or in a local analysis method that does not increase the rank of the matrix using a Schur product), as shown in Tippett et al. (2003), the Sherman Woodbury update is sufficient for an unlocalized matrix as the rank of the matrix is at most the number of ensemble members (Godinez and Moulton 2012). However, the fundamental shift to high-rank matrices requires additional effort to correct.

Several strategies have been proposed to handle this observation-ordering dependence within a serial filter. Bishop et al. (2015) proposes the Consistent Hybrid Ensemble Filter (CHEF) with local analysis and perturbed observations that will ensure the analysis is consistent and does not depend on the order of assimilation. Kotsuki et al. (2017) presents a study of observation ordering with a Lorenz-96 model and investigates rules for observation assimilation ordering to minimize analysis forecast error. The method of correcting sample correlation described in Anderson (2012) has also been used to reduce the dependence of observation ordering in a serial filter (J. Anderson 2017, personal communication).

Extending upon these works, as an alternative to attempting to apply and update the high-rank localized matrices serially in a consistent way, we propose assimilating all observations within the assimilation window in a single pass as a potential alternative. In other words, we do not utilize the single observation processing strategy normally employed for serial filter solutions and instead solve the ESRF equations directly. This is done by dividing the necessary matrix operators across the set of processing elements in a "top-down" fashion as opposed to the "bottom-up" approach of local analysis. This method was utilized in Steward et al. (2017) (hereafter S17) to provide a global, "all-at-once," parallel, direct solution of the covariance-localized ESRF equations. Note that "all-at-once" here is used to refer to assimilating all observations that the serial filter would assimilate one-by-one but not all observations within all assimilation windows at once, i.e. the method in S17 as well as the one presented below are both sequential filters in that batches of observations can also be assimilated. The benefit of this approach is that the analysis consistently applies the high-rank covariance-localized matrices and, as a result, does not depend on the order of observations. It provides a solution to the ESRF equations with a proven error bounds that can be used as a benchmark against other methodologies.

The cost of this approach is that a product with the entire full-rank matrix inverse (which also requires a square-root term) of the forward observation error covariance is required. S17 solves for eigenvalues and eigenvectors of the observation covariance matrix and uses the ESRF matrix function "scalarized" on the eigenvalues to find the required matrix inverses and products. As eigenpairs are extremely convenient for mathematical analysis, the approach in S17 also includes an error bounds related to the smallest eigenvalue used. The final analysis is also shown not to depend on the ordering of observations. This error-bounded method, which directly solves the ESRF equations, is therefore a highly accurate solution to the ESRF equations known to be the minimum variance solution to the data assimilation problem.

However, as predicted by theory and shown in this work, while the method described in S17 is accurate to within a configurable tolerance, it is impractical for large numbers of observations due to the nature of the eigenproblem, where for general matrices finding a large number of eigenpairs scales as $O(n^3)$ for a matrix of size $n \times n$ (Golub and Van Loan 1996). n is the number of quality-controlled observations in this case. This paper extends S17 to take advantage of recent improvements in the theory and computation of matrix functions to transform the problem of solving the difficult inverse and square-root portion of the ESRF equations into to computing matrix-vector products that are used to build up a Krylov subspace and, through a library call, applying the matrix function directly to a small dense matrix. This small dense matrix represents the compression of the larger localized forward-observation covariance matrix onto the reduced-order Krylov subspace basis.

As we show below, this matrix function method gives results that are practically identical to the error-bounded methodology of S17 but is much more computationally efficient. As only a matrix-vector product with the observation covariance matrix is required, this matrix function approach is well-suited for a matrix-free implementation where the covariance matrix is not explicitly formed. This method is also amenable to hybrid covariance models using both ensemble and climatological covariances.

We implement the matrix function method and compare the performance results with both S17 as well as the parallel augmented-state method of Anderson and Collins (2007), hereafter referred to as AC07. As a proof-of-concept application, we test this method on the difficult, highly-nonlinear case of first cycle tropical cyclone (TC) data assimilation. In this case, the background ensemble can contain position errors of features and the posterior analysis increment can be large (e.g. Chang et al. 2014). As we show, the order-dependence issue of a serial filter is non-trivial in this case. In order to demonstrate the unique properties of our new method, we investigate TC assimilation with a long covariance length-scale that would be impractical for local analysis methods. As we show, the matrix function method is roughly comparable in terms of wall-time performance to AC07 and far superior to S17. The analysis results do not depend on observation ordering like S17 but contrary to AC07. However, our results demonstrate the memory scaling of the matrix function method is inferior to AC07, and suggest that matrix-free methods would be required to scale this method to the order of millions of observations at once.

This paper is organized as follows. Section 2 summarizes S17 in order to build upon it. In section 3, the eigenpair computation of S17 is replaced with a much more efficient matrix function based approach that uses a basis for the Krylov space to compress the forward observation covariance matrix and apply the covariance-localized ESRF matrix functions to this reduced-order matrix. Section 4 summarizes AC07. Section 5 presents numerical results of the matrix function approach and a performance comparison to S17 and AC07. Finally, section 6 presents conclusions and a discussion.

2. Eigenvalue/eigenvector solution of S17

In this section, we briefly review S17 in order to introduce the new matrix function method that extends it. Given an ensemble $\mathbf{X}_f$ of a previous forecast, the updated analysis to the ensemble mean $\bar{\mathbf{x}}_f$ of size $N_{\text{state}} \times 1$ and ensemble perturbations $\mathbf{X}'_f$ of size $N_{\text{state}} \times N_{\text{ens}}$, the square-root Ensemble Kalman filter without perturbed observations (Whitaker and Hamill 2002) is:

$$\begin{aligned} \bar{\mathbf{x}}_a &= \bar{\mathbf{x}}_f + \mathbf{K}\left(\mathbf{y} - \overline{H(\mathbf{X}_f)}\right), \\ \mathbf{X}'_a &= \mathbf{X}'_f + \tilde{\mathbf{K}}\left(\mathbf{0} - H\mathbf{X}\right), \end{aligned} \tag{1}$$

where $\mathbf{y}$ ($N_{\text{obs}} \times 1$) are the observations, $\overline{H(\mathbf{X}_f)}$ ($N_{\text{obs}} \times 1$) is the mean of the forward-calculated observation operators, and $H\mathbf{X}_{i,j} = h_i\left(\mathbf{X}_f^{(j)}\right) - \overline{h_i\left(\mathbf{X}_f\right)}$ is the mean-subtracted i^{th} observation operator acting on the j^{th} ensemble member $\mathbf{X}_f^{(j)}$. $H\mathbf{X}$ is $N_{\text{obs}} \times N_{\text{ens}}$ (as is $\mathbf{0}$, a matrix filled with zeros). The traditional Kalman gain, $\mathbf{K}$ ($N_{\text{state}} \times N_{\text{obs}}$), is

$$\mathbf{K} = \mathbf{C}_{\mathbf{x},H\mathbf{x}}\,\mathbf{D}^{-1}, \tag{2}$$

where $\mathbf{C}_{\mathbf{x},H\mathbf{x}} = \operatorname{cov}\left(\mathbf{x}_f, H\left(\mathbf{x}_f\right)\right)$ is the localized covariance between $\mathbf{x}_f$ (an $N_{\text{state}} \times 1$ random variable representing the previous forecast) and $H(\mathbf{x}_f)$ (the observation operator acting on this random variable). $\mathbf{D} = \mathbf{C}_{H\mathbf{x},H\mathbf{x}} + \mathbf{R}$ for $\mathbf{C}_{H\mathbf{x},H\mathbf{x}} = \operatorname{cov}\left(H\left(\mathbf{x}_f\right), H\left(\mathbf{x}_f\right)\right)$, the localized forward observation covariance, and $\mathbf{R}$ is the observation error covariance $\operatorname{cov}\left(\mathbf{y}_t - H(\mathbf{x}_f)\right)$ for a random variable $\mathbf{y}_t$ representing the true observations without observation noise.

$\tilde{\mathbf{K}}$ ($N_{\text{state}} \times N_{\text{obs}}$), the correction from using non-perturbed observations, is

$$\tilde{\mathbf{K}} = \mathbf{C}_{\mathbf{x},H\mathbf{x}}\,\mathbf{D}^{-1/2}\left(\sqrt{\mathbf{D}} + \sqrt{\mathbf{R}}\right)^{-1}. \tag{3}$$

As detailed in S17, the covariance matrices we consider can include localized-ensemble based correlations in observation space and/or variational-style model-space localization. For observation-space localization, a component-wise multiplication $\circ$ between two matrices is used as

$$\mathbf{C}^{\text{obs}}_{H\mathbf{x},H\mathbf{x}} = \rho_{\mathbf{y},\mathbf{y}} \circ \mathbf{Q}_{H\mathbf{x},H\mathbf{x}} \quad (4)$$

where $\rho_{\mathbf{y},\mathbf{y}}$ is the localization matrix arising from a localization function (Gaspari and Cohn 1999) ℓ such that

$$(\rho_{\mathbf{y},\mathbf{y}})_{i,j} = \ell(d_{i,j}|L_{i,j}) \quad (5)$$

where $d_{i,j}$ is the distance between the location of the i^{th} and j^{th} observation, and $L_{i,j}$ is the characteristic length scale for the localization function ℓ. $\mathbf{Q}_{H\mathbf{x},H\mathbf{x}}$ is the sample covariance matrix

$$\mathbf{Q}_{H\mathbf{x},H\mathbf{x}} = \frac{H\mathbf{X}\,(H\mathbf{X})^{\text{T}}}{N_{\text{ens}} - 1}. \quad (6)$$

Likewise, the observation-space localized model and observation cross-covariance is given by

$$\mathbf{C}^{\text{obs}}_{\mathbf{x},H\mathbf{x}} = \rho_{\mathbf{x},\mathbf{y}} \circ \mathbf{Q}_{\mathbf{x},H\mathbf{x}} \quad (7)$$

for

$$(\rho_{\mathbf{x},\mathbf{y}})_{i,j} = \ell(d_{i,j}|L_{i,j}) \quad (8)$$

where $d_{i,j}$ is the distance between the location of the model state i and observation j with the same localization function as equation (5), and

$$\mathbf{Q}_{\mathbf{x},H\mathbf{x}} = \frac{\mathbf{X}'_f\,(H\mathbf{X})^{\text{T}}}{N_{\text{ens}} - 1} \quad (9)$$

As noted in Campbell et al. (2010), integrated observations such as satellite scans do not have a particular vertical location to ascribe. In these cases, model-space localization is more applicable. For model-space localization, the observation operator tangent-linear $\mathbf{H}$ and adjoint $\mathbf{H}^{\text{T}}$ are applied to the localized model covariance as

$$\mathbf{C}^{\text{model}}_{H\mathbf{x},H\mathbf{x}} = \mathbf{H}\left(\rho_{\mathbf{x},\mathbf{x}} \circ \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)\mathbf{H}^{\mathrm{T}} \tag{10}$$

where

$$\mathbf{Q}_{\mathbf{x},\mathbf{x}} = \frac{\mathbf{X}'_f\left(\mathbf{X}'_f\right)^{\mathrm{T}}}{N_{\text{ens}}-1} \tag{11}$$

for the ensemble perturbations $\mathbf{X}'_f$, and

$$\left(\rho_{\mathbf{x},\mathbf{x}}\right)_{i,j} = \ell(d_{i,j}|L_{i,j}) \tag{12}$$

where $d_{i,j}$ is the distance between the location of two model states i and j with the same localization function as equation (5). Equation (7) is changed analogously as

$$\mathbf{C}^{\text{model}}_{\mathbf{x},H\mathbf{x}} = \left(\rho_{\mathbf{x},\mathbf{x}} \circ \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)\mathbf{H}^{\mathrm{T}} \tag{13}$$

Note that all of these localized matrices are sparse, and zero elements (i.e. the correlations farther than the specified localization distance) are not stored in memory or computed. Thus, for example, only those elements of $\mathbf{Q}_{H\mathbf{x},H\mathbf{x}}$ that will be non-zero after localization are calculated. Furthermore, the full model-space matrix $\mathbf{Q}_{\mathbf{x},\mathbf{x}}$ will never be explicitly formed due to its prohibitively large size. See S17 for more detail.

As we will allow for full-rank matrices, our method is compatible with either of these localization methods, a linear combination of the two, or any other "reasonable" modeled covariance between $H\mathbf{x}$ and $H\mathbf{x}$ and $\mathbf{x}$ and $H\mathbf{x}$, which we denote in general $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ and $\mathbf{C}_{\mathbf{x},H\mathbf{x}}$. Note that in this work we only present results for the observation-based localization of equations (4) and (7), however.

We now return to solving equation (1). Both S17 and the matrix function approach utilize a pre-processing step of a transformation first introduced in Bishop et al. (2001) to whiten the observations as $\mathbf{y} = \mathbf{R}_{\text{old}}^{-1/2}\mathbf{y}_{\text{old}}$, where the "old" subscript represents the untransformed observations.

The observation operator is also scaled as $H(\mathbf{x}) = \mathbf{R}_{\text{old}}^{-1/2} H_{\text{old}}(\mathbf{x})$. As a result of this pre-processing transformation, the $\mathbf{R}$ matrix is now identity, which makes equation (3) much easier to solve. For the diagonal observation error matrix $\mathbf{R}_{\text{old}}$ typically used in data assimilation (which assumes uncorrelated observation errors), multiplying by $\mathbf{R}_{\text{old}}^{-1/2}$ is equivalent to dividing each observation by the standard deviation of the observation error, and for non-diagonal $\mathbf{R}_{\text{old}}$, this transformation removes that off-diagonal correlation using principal components.

As $\mathbf{D}_{\text{new}} = \mathbf{C}_{H\mathbf{x},H\mathbf{x}} + \mathbf{I}$ by this transformation (note that we drop the "new" subscript in what follows as it could be applied to virtually all matrices; i.e. we write $\mathbf{D}_{\text{new}}$ as $\mathbf{D}$ in a slight abuse of notation), this leads to

$$\tilde{\mathbf{K}} = \mathbf{C}_{\mathbf{x},H\mathbf{x}}\,\mathbf{M}^{-1} \tag{14}$$

for $\mathbf{M} = \mathbf{D} + \sqrt{\mathbf{D}}$. Let $\lambda_i, \mathbf{v}_i$ denote the i^{th} eigenpair of $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$. Then

$$\mathbf{M}\mathbf{v}_i = \mathbf{C}_{H\mathbf{x},H\mathbf{x}}\,\mathbf{v}_i + \mathbf{v}_i + (\mathbf{C}_{H\mathbf{x},H\mathbf{x}} + \mathbf{I})^{1/2}\,\mathbf{v}_i\,. \tag{15}$$

As shown in S17, we have

$$\mathbf{M}\mathbf{v}_i = \left(\lambda_i + 1 + (\lambda_i + 1)^{1/2}\right)\mathbf{v}_i\,. \tag{16}$$

Therefore,

$$\mathbf{M}^{-1}\mathbf{v}_i = \lambda_i' \mathbf{v}_i \tag{17}$$

for

$$\lambda_i' = \frac{1}{\lambda_i + 1 + (\lambda_i + 1)^{1/2}}\,. \tag{18}$$

We find the largest r eigenvalues and corresponding eigenvectors of $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$, where r is chosen such that $\lambda_{r+1} \leq \varepsilon_\lambda$ for some small constant ε_λ, and we can therefore solve

$$\mathbf{M}^{-1}(\mathbf{0} - H\mathbf{X})_j \approx \sum_{i=1}^{r} \lambda_i' \alpha_{i,j} \mathbf{v}_i \tag{19}$$

for $\alpha_{i,j} = -\mathbf{v}_i^{\mathrm{T}} H\mathbf{X}_j$. An error bound on this approximation related to ε_λ is proved in S17.

Similarly, for the mean update:

$$\mathbf{D}^{-1}(\mathbf{y}-\overline{H(\mathbf{X}_f)}) \approx \sum_{i=1}^{r} \frac{\beta_i}{\lambda_i+1}\mathbf{v}_i\,, \tag{20}$$

where $\beta_i = \mathbf{v}_i^{\mathrm{T}}(\mathbf{y}-\overline{H(\mathbf{X}_f)})$.

The $N_{\mathrm{obs}} \times N_{\mathrm{ens}}+1$ matrix $(\mathbf{E}|\mathbf{g})$, where

$$\mathbf{E}_j = \sum_{i=1}^{r} \lambda_i' \alpha_{i,j}\mathbf{v}_i \tag{21}$$

and

$$\mathbf{g} = \sum_{i=1}^{r} \frac{\beta_i}{\lambda_i+1}\mathbf{v}_i\,, \tag{22}$$

is then distributed to all processing elements. The remaining Kalman gain from equation (1) only requires multiplication with $\mathbf{C}_{\mathbf{x},H\mathbf{x}}$, which can proceed in an embarrassingly parallel fashion. This makes an efficient parallel method that only requires the eigenpairs of the $N_{\mathrm{obs}} \times N_{\mathrm{obs}}$, sparse, positive semi-definite symmetric matrix $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$. The Scalable Library for Eigenproblem Computation (SLEPc, Hernandez et al. 2005), which is built upon the Portable Extensible Toolkit for Scientific Computing (PETSc, Balay et al. 1997, 2016, 2017), is used to solve this eigenproblem using sparse matrices in a manner that scales well as a function of the number of processors, as shown in S17.

3. New matrix function approach

We first note that while S17 evaluates the largest r eigenpairs of $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ in order to solve equation (1), only those eigenvectors i such that $\alpha_{i,j} \neq 0$ for all j and $\beta_i \neq 0$ are required. This suggests a more efficient solution that does not require all eigenpairs. In this section we develop such a solution that requires only the matrix-vector product $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}\mathbf{b}$ for some vector $\mathbf{b}$ to compute a reduced-order, accurate basis for representation of the ESRF matrix functions.

In addition to solving the eigenproblem, SLEPc can also evaluate the action of a matrix function on a vector, $\mathbf{z} = f(\mathbf{A})\mathbf{b}$ where $\mathbf{z}$ and $\mathbf{b}$ are vectors, $\mathbf{A}$ is a matrix, and f is a matrix function in the sense given in Higham (2008). In the case of the mean $\mathbf{K}$ in equation (2) given above

$$f_1(\mathbf{D}) = \mathbf{D}^{-1}, \tag{23}$$

while for $\tilde{\mathbf{K}}$ in equation (14),

$$f_2(\mathbf{D}) = (\mathbf{D} + \sqrt{\mathbf{D}})^{-1}. \tag{24}$$

Recall that $\mathbf{D} = \mathbf{C}_{H\mathbf{x},H\mathbf{x}} + \mathbf{I}$. Also note that f_1 involves the standard linear system of equations $\mathbf{D}\mathbf{x} = \mathbf{b}$, solving for $\mathbf{x}$, which is normally handled by other methods; in this work, we test using the matrix function approach for both the mean and the perturbations.

The matrix function solvers in SLEPc are based on Krylov subspace methods (Higham 2008, Ch. 13). Earlier works using Krylov subspace methods to approximate matrix functions include Van Der Vorst (1987), Saad (1992) and Hochbruck and Lubich (1997). These methods are appropriate for the case of our large, high-rank matrix $\mathbf{D}$ as they compute the result $\mathbf{z}$ without explicitly building the matrix $f(\mathbf{D})$. The calculation of $f(\mathbf{D})\mathbf{b}$ proceeds in a manner similar to the Arnoldi method (Arnoldi 1951) for finding eigenpairs. At the first step, $\mathbf{V}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|_2}$, and at step m, given an $N_{\text{obs}} \times (m-1)$ orthonormal basis $\mathbf{V}_{m-1}$ of the Krylov subspace $\mathscr{K}_{m-1}(\mathbf{D},\mathbf{b}) = \text{span}\left\{\mathbf{b}, \mathbf{D}\mathbf{b}, \mathbf{D}^2\mathbf{b}, \ldots, \mathbf{D}^{m-2}\mathbf{b}\right\}$, we seek the orthonormal basis $\mathbf{V}_m$ that spans $\mathscr{K}_m(\mathbf{D},\mathbf{b})$. This is done by the Arnoldi relation $\mathbf{D}\mathbf{V}_{m-1} = \mathbf{V}_{m-1}\mathbf{H}_{m-1} + h_{m,m-1}\mathbf{v}_m\mathbf{e}_{m-1}^T$, where $\mathbf{H}_{m-1}$ is an $(m-1)\times(m-1)$ upper Hessenberg matrix that contains the values of the projections of $\mathbf{D}$ onto the basis $\mathbf{V}_{m-1}$, $\mathbf{v}_m$ is the m^{th} column to be added to $\mathbf{V}_m$ this iteration, and $h_{m,m-1}$ is the $(m, m-1)$ entry in the $\mathbf{H}_m$ matrix. $\mathbf{e}_{m-1}$ is the $m-1$ unit coordinate vector, so $h_{m,m-1}\mathbf{v}_m\mathbf{e}_{m-1}^T$ is the $N_{\text{obs}} \times (m-1)$ zero matrix except column $m-1$ which is $h_{m,m-1}\mathbf{v}_m$. Once $\mathbf{V}_m$ is found, the approximation of $\mathbf{z}$ can be computed as

$$\tilde{\mathbf{z}}_m = \beta \mathbf{V}_m f(\mathbf{H}_m)\mathbf{e}_1 \tag{25}$$

where $\beta = \|\mathbf{b}\|_2$. $\mathbf{e}_1$ is the first coordinate vector, so right multiplying by it gives the first column of $\beta \mathbf{V}_m f(\mathbf{H}_m)$ in equation (25). Note that $\mathbf{b} = \beta \mathbf{V}_m \mathbf{e}_1$. In addition, note that $\mathbf{H}_m$ represents the compression of $\mathbf{D}$ onto $\mathscr{K}_m(\mathbf{D},\mathbf{b})$ with respect to the basis $\mathbf{V}_m$. Hence, the problem of computing the function of a large matrix $\mathbf{D}$ of order N_{obs} is reduced to computing the function of a small matrix $\mathbf{H}_m$ of order m with $m \ll N_{\text{obs}}$. For the latter task, we can employ algorithms for dense matrices as discussed below.

Note that in the above description, the Arnoldi process requires a numerically stabilized Gram-Schmidt process to orthonormalize the basis vectors in a way that the final result is not overly affected by numerical noise. Furthermore, the parallelization of this stabilized process requires careful implementation to avoid negatively impacting performance by creating bottlenecks. Thus, the relatively straight-forward (conceptually) Gram-Schmidt process becomes rather complex when implemented in a parallel setting as discussed in Björck (1994) and Frayssé et al. (1998). SLEPc utilizes an efficient parallel version of the Iterated Classical Gram-Schmidt (ICGS) in the Arnoldi process that does not require global communication but maintains numerical stability. As in the high-level description given above, the resulting projections in the ICGS process onto the previous basis vectors are stored in the $\mathbf{H}_m$ matrix. For more details on the orthogonalization process in SLEPc, see Hernandez et al. (2007).

The m parameter is of paramount importance for this method. If m is too small the Krylov subspace will not contain enough information to build an accurate approximation. On the other hand, if m is too large, the memory requirements for storing $\mathbf{V}_m$ (as well as the computational cost) will be prohibitive. For this reason, SLEPc implements a restarted variant of the method, where m is prescribed to a fixed value; here we use $m = 150$, which as shown below is based on testing for our particular application. When the subspace reaches this size, a restart is carried out by keeping part of the data computed so far and discarding unnecessary information. Investigation into restarting matrix function iterations is still an area of active research (Afanasjew et al. 2008; Eiermann et al. 2011; Frommer et al. 2017). SLEPc implements the Eiermann-Ernst restart (Eiermann and Ernst 2006), in which only the last basis vector $\mathbf{v}_{m+1}$ is kept (in order to continue the Arnoldi recurrence) along with the matrix $\mathbf{H}_m$ that is "glued" together with the previous ones. After k restarts, the matrix used in the approximation (25) has the form

$$\mathbf{H}_{km} = \begin{bmatrix} \mathbf{H}_{(k-1)m} & \mathbf{0}_m \\ h^{(k-1)}_{m+1,m}\mathbf{e}_1\mathbf{e}^T_{(k-1)m} & \mathbf{H}^{(k)}_m \end{bmatrix}, \tag{26}$$

where $\mathbf{H}^{(k)}_m$ is the matrix computed by the Arnoldi method in the k^{th} restart. Note that in the Eiermann-Ernst restart, the glued matrix (26) is not used directly in (25) because $\mathbf{H}_{km}$ has size $km \times km$ but $\mathbf{V}_m$ has only m columns. Therefore, only the last m components of the vector $f(\mathbf{H}_{km})\mathbf{e}_1$ are used in (25) to give a correction to be added to the approximation available in the previous restart. This correction is given by $\tilde{\mathbf{z}}^{(k)} = \tilde{\mathbf{z}}^{(k-1)} + \mathbf{c}^{(k)}$, where

$$\mathbf{c}^{(k)} = \beta \mathbf{V}^{(k)}_m [\mathbf{0}, \mathbf{I}_m] f(\mathbf{H}_{km})\mathbf{e}_1 \,, \tag{27}$$

and $\mathbf{V}^{(k)}_m$ is the basis computed in the last restart. Equations (25) through (27) are implemented in a numerically efficient way in SLEPc.

SLEPc bases the stopping criterion on the norm of the correction, i.e. restarting continues until $\|\mathbf{c}^{(k)}\|_2 < \beta \cdot \varepsilon_{\text{tol}}$ for some user-defined ε_{tol} (10^{-8} by default for 8-byte floating point precision). As noted in Eiermann and Ernst (2006), the Arnoldi method converges rapidly with superlinear behavior for smooth functions. The convergence behavior when including restarting is presented in Afanasjew et al. (2008) for a related method.

In this work, we are interested in solving $\mathbf{g} = f_1(\mathbf{D})(\mathbf{y} - \overline{H(\mathbf{X}_f)})$ to replace equation (22) and $\mathbf{E}_j = f_2(\mathbf{D})(\mathbf{0} - H\mathbf{X})_j$ to replace equation (21) for $j = 1, \ldots, N_{\mathrm{ens}}$. Applying the method described above leads to the evaluation of $f_1(\mathbf{H}_1)$ and $f_2(\mathbf{H}_2)$ explicitly for small dense matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ of the form in eq. (26). Note that these matrices are not symmetric even though $\mathbf{D}$ is symmetric, and also note that the matrices grow at each restart of the Krylov method.

SLEPc allows flexibility in the definition of functions by combining two simpler functions. In our case, we define $f_1(\cdot)$ as the reciprocal of the identity function and $f_2(\cdot)$ as the reciprocal of another function, which in turn is defined as the sum of two functions (identity and the square root). All these sub-functions can be evaluated easily except the matrix square root. For this SLEPc implements a reduction to (real) Schur form followed by a block version of a Schur algorithm (Higham 1987; Deadman et al. 2012).

Note that only the matrix action $\mathbf{Db}$ is required in this algorithm, allowing for matrix-free implementations. This could be potentially useful for defining matrix-vector products using the "modulation product" defined in Bishop and Hodyss (2009) or for variational-style covariances that use Fast Fourier Transforms (FFT) to define the action of a circulant covariance matrix. Hybrid methods are also possible; as long as the action of the covariance $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ as well as $\mathbf{C}_{\mathbf{x},H\mathbf{x}}$ can be applied, any such modeled covariance can be imposed on the analysis through the ESRF equations through this approach.

4. Serial augmented-state filter of AC07

In order to compare the performance of our new matrix function approach to an existing method, we briefly summarize the method of AC07 here. AC07 details a highly scalable approach to solving the ESRF equations in serial that is provably identical to the global solution with linear observation operators and without covariance localization. With covariance localization, however, the results will depend upon the ordering of observations as discussed above, although to what extent this difference will impact ensemble NWP forecasts has not yet been explored.

AC07 describes an algorithm that loops over each observation in serial. Each observation is owned by a particular processing element. For each observation n, the owner of that observation broadcasts the observation details (including the observation location, ensemble forward-calculated values $h_n(\mathbf{x}_j)$ for $j = 1, \ldots, N_{\text{ens}}$, and QC status) to the other processing elements, which then each process the observation in parallel. An important innovation of AC07 is the treatment of observations themselves as part of the augmented state vector. In other words, just as water vapor, temperature, and other geophysical variables are updated by the Kalman filter equations, the observations (which are assumed to have a particular location in space) are also updated during the assimilation process. Thus the n^{th} observation that is broadcast by the owner processing element will have been potentially updated by observations 1 through $n-1$. This saves the computational expense of having to communicate in order to recompute the observation operators.

A scalar form of the ESRF equations (1) is used to efficiently update all of the covariance localized state points and observations. The mean of each state i is updated as

$$\bar{\mathbf{x}}_i = \bar{\mathbf{x}}_i + k_{i,n}\left(\mathbf{y}_n - (\overline{H(\mathbf{X}_f)})_n\right) \tag{28}$$

for the Kalman gain $k_{i,n}$ from equation (2) scalarized for point i for observation n as

$$k_{i,n} = \frac{\rho_{i,n}}{d_n} \frac{1}{N_{\text{ens}} - 1} \sum_{j=1}^{N_{\text{ens}}} (\mathbf{X}'_f)_{i,j} (H\mathbf{X})_{n,j}. \tag{29}$$

Here,

$$d_n = \frac{1}{N_{\text{ens}} - 1} \sum_{j=1}^{N_{\text{ens}}} (H\mathbf{X})^2_{n,j} + \mathbf{R}_{n,n}, \tag{30}$$

where $\mathbf{R}_{n,n}$ ($\mathbf{R}$ is assumed diagonal) is the observation error variance of the n^{th} observation, and $\rho_{i,n}$ is the localization factor between the state point i and observation n, i.e. it corresponds to the (i,n) component of the $\rho_{\mathbf{x},\mathbf{y}}$ matrix in equation (8), although this matrix is not formed in this implementation.

Similarly, given the scalar square-root correction

$$\beta_n = \frac{1}{1+\sqrt{r_d}}, \tag{31}$$

where

$$r_d = \frac{\mathbf{R}_{n,n}}{d_n}, \tag{32}$$

the j^{th} ensemble perturbation at state point i is updated as

$$\mathbf{X}'_{i,j} = \mathbf{X}'_{i,j} + \beta_n k_{i,n} \left(0 - (H\mathbf{X})_{n,j}\right). \tag{33}$$

Note that the analogous equations are used to update the approximations of the forward observation mean $(\overline{H(\mathbf{X}_f)})_k$ and perturbations $(H\mathbf{X})_{k,j}$ for $k = n+1$ to N_{obs}, i.e. the remaining unassimilated forward observations are treated as part of the augmented state vector.

5. Numerical results

The implementation described in section 3 was used to replace the computation of $(\mathbf{E}|\mathbf{g})$ from equations (21) and (22) from S17, retaining the remaining components. For comparison, the serial method of AC07 was implemented and tested as well. To ensure consistent comparisons, an object-oriented approach was incorporated in the Hurricane Ensemble Data Assimilation System (HEDAS, Aksoy et al. 2012, 2013; Aksoy 2013; Vukicevic et al. 2013; Aberson et al. 2015) to maintain consistency in observation processing, quality control, and disk input/output among all three implementations. Only the filter aspect differs.

All timings were tested on the NOAA Jet supercomputing system xjet installed in 2015/2016 where each node has 24 cores with a 2.3 GHz Intel Haswell CPU and 2.66 GB RAM connected via FDR Infiniband. As a proof-of-concept for this method, we ran two experiments, each with

30 Hurricane WRF (Gopalakrishnan et al. 2010, HWRF) ensemble members, using the Hurricane Edouard (2014) study described in Christophersen et al. (2017). Both of these experiments use quality-controlled observations from sources including satellite retrievals and the NASA AV6 Global Hawk 20140916GH Storm Survey mission (Zawislak et al. 2016; Rogers et al. 2016; Christophersen et al. 2017).

The first experiment, to illustrate the performance on a relevant single cycle as in Christophersen et al. (2017), uses HWRF to spin up 30 GFS ensemble members initialized at 2014-09-16 12:00 UTC for 4 hours, then assimilates 15.2K quality-controlled observations from this set at 2014-09-16 16:00 UTC $\pm$ 30 minutes using the HEDAS system. The localization length-scale was set to $L = 240$ as $c = L/2$ from equation 4.10 of Gaspari and Cohn (1999) as described in S17. Figure 1 shows the analyzed water vapor field at level 20 (out of 60) for the EPS, MFN, and serial implementation of AC07. Ten different random observation orderings were assimilated. The mean and standard deviation of the ten different AC07 analyses are shown in fig. 1a) and 1b). As shown, the standard deviation of these different orderings can reach up to approximately 1.5 g kg^{-1}. The same 10 random orderings were assimilated with the MFN solution as shown in fig. 1c) and 1d). Each time, the MFN analysis was identical to within 10^{-7}; the standard deviation is less than 10^{-7} ("zero") as well. For comparison, the absolute difference between the average serial analysis and the EPS analysis is shown in fig. 1e), which as shown is greater than 2 g kg^{-1} in places. The absolute difference between the MFN and EPS solution is shown in fig. 1f), which is also "zero."

To emphasize the order independence issue, figure 2 shows the assimilation of the first two random observation orderings assimilated in figure 1 (order 1 and order 2). No effort was made to maximize this difference for AC07 – the first two random orderings were chosen – but likewise no attempt was made to minimize forecast impact in AC07 by optimizing the ordering as in Kotsuki et al. (2017). The differences at this level reach up to 3.5 g kg^{-1}. The root-mean-squared difference of the entire domain at this level was approximately 0.5 g kg^{-1}. However, the MFN analyzed solutions with different orderings were found to be identical to within 10^{-8}. A similar tolerance was found by comparing the MFN and EPS solutions.

Figure 3a) shows the level 20 water vapor standard deviation (across the ensemble) of the prior ensemble perturbations $\mathbf{X}'_f$, while the standard deviation of the MFN posterior perturbations $\mathbf{X}'_a$ with orderings 1 and 2 (which are numerically equivalent up to single precision) is shown in fig. 3b). Figure 3c) and 3d) show the standard deviation of $\mathbf{X}'_a$ at this level for the ordering 1 and 2, respectively, with the AC07 filter. Figure 3e) shows the two standard deviations differ by up to 0.1 g kg^{-1}, while the difference between the AC07 order 1 $\mathbf{X}'_a$ and the EPS solution is up to 0.35 g kg^{-1}. As in the mean, the MFN perturbations and the EPS perturbations are identical to within 10^{-7}.

As shown in figures 1 through 3, the differences in the $\bar{\mathbf{x}}_a$ analysis with random orderings using the AC07 filter are large enough that they are comparable to the posterior covariance in certain locations. This is likely due to the highly non-linear nature of the first-cycle tropical cyclone data assimilation problem. In this application, flights are used as observing platforms to narrow the inner core uncertainty as shown in figure 3. The first cycle background contains ensemble members with simulated tropical cyclones with features centered at different locations, leading to large analysis updates. The main area of uncertainty in the AC07 analyses is actually outside of the inner core in the south-west quadrant near an area of dry air inflow. As shown, over the different serial courses of assimilation the order-dependent error standard deviation of this region can grow to be roughly equivalent in magnitude to the posterior covariance. The matrix function approach, however, is order independent and therefore removes this source of error and is thus more numerically consistent with the eigenpair-based solution to the ESRF equations.

Having established that in this case the matrix function solution is numerically similar to the EPS method, which has a proven error bounds, we now turn our attention to the computational performance of the new method. For this purpose we use a second experimental setup that combines the observations at all times that fall within the same domain as the first experiment. This leads to up to 35,420 quality-controlled observations that can be used for performance testing.

Keeping 1/2 of the total number of observations from all cycles fixed at 17.7K, the scaling as a function of number of cores is shown in figure 4. The matrix function method scales nearly linearly as a function of the number of processing elements as in S17, but overall the wall time remains bound by I/O time.

As a function of the number of observations the MFN implementation scales much better than the eigenproblem-based solution (EPS) as shown in figure 5, where the number of processing elements is fixed at 386, $L = 240$ for the correlation length-scale, and the number of observations vary. As discussed in S17, $L = 240$ leads to points across more than half of the domain being correlated which in turn leads to a relatively dense, nearly full-rank matrix. As predicted by theory, the EPS solution appears to scale as the cube of the number of observations. However, the MFN approach apparently scales linearly. Times for the EPS solution longer than 45 minutes are not shown. With 17.7k observations on 386 processing elements, the EPS solution took 41 minutes and 28 seconds to complete from start to finish (including expensive disk reading and writing), while the MFN solution took only 16 minutes and 42 seconds. The MFN solution continues to scale well even at 35.4K observations, completing in 30 minutes and 45 seconds, which is still more than 10 minutes faster than the EPS solution with half as many observations. Therefore, as shown, the MFN approach scales much better as function of the number of observations than the EPS solution.

The MFN solution is also roughly comparable to the AC07 solution in terms of wall time. While the MFN approach is actually slightly faster for small amounts of observations, for the largest number of observations tested (35.4K observations) the serial filter is faster with a wall time of 28 minutes 24 seconds as opposed to 30 minutes 45 seconds. However, the wall-time differences are small enough that the observation order independence of MFN apparently makes it competitive with AC07 for these numbers of observations. This is somewhat surprising as the only communication used by the AC07 filter is to broadcast observations, while distributed matrix multiplications are required by the MFN approach. However, the MFN approach has the potential benefit that it does not serially iterate over the observations, but instead can process all observations in parallel.

The number of matrix multiplications, and hence the overall timing of the matrix function solution, is directly related to the number of restarts and m, the maximum basis size before restarting. Increasing m leads to fewer restarts but requires additional memory and dense matrix processing time. The number of Eiermann-Ernst restarts necessary for convergence with $m = 150$ as used in our study ranged from 1 for the smallest number of observations (2,760) to 2 for the largest number of observations (35,420). The SLEPc error estimate at the end of each restart iteration for the smallest number of observations was on the order of 10^{-2} for $k = 0$ and 10^{-15} for $k = 1$, while for the largest the error was on the order of 10^{-2} for $k = 0$, 10^{-8} for $k = 1$, and 10^{-13} for $k = 2$. It appears the number of restarts grows very weakly with N_{obs}.

Table 1 shows the time necessary to solve the matrix function portion of the ESRF equations per ensemble member with $L = 240$ for the 17.7k observation case as a function of varying the m parameter. As shown, m less than 100 requires an excessive amount of restarts and total matrix product evaluations; for m greater than 100, the overall performance is dependent upon the exact number of matrix product evaluations required to reach the numerical accuracy of $\varepsilon_{\mathrm{tol}} = 10^{-8}$. For this case, $m = 125$ requires the fewest number of matrix-product evaluations, which is highly correlated with the total amount of MFN solve time. Table 2 shows the same results with the localization length-scale $L = 60$. In this case, $m = 150$ gives the optimal results. The best particular value of m therefore depends upon the factorization of the total number of evaluations required. m larger than 100 is recommended to avoid excessive restarting, and m less than 200 is recommended due to the expense of dense matrix evaluations. We choose $m = 150$ to split the difference.

The scaling of memory usage on 386 xjet processors as a function of number of observations is shown in fig. 6. As shown, and as expected by theory, the EPS solution memory usage scales cubically as a function of the number of observations. The serial filter of AC07 apparently scales linearly as it only processes a single observation at once. The MFN solution, which currently stores the entire sparse $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ matrix in memory, scales better than S17 but apparently worse than linearly. This is because with $L = 240$ the $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ matrix is relatively dense. For a dense matrix, the memory requirements would be quadratic, while for a sparse matrix the memory requirements would be closer to linear. The memory scaling here is consistent with a factor somewhere in between quadratic and linear. Note, however, that the expense here is related to the representation of $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ and not directly to the MFN approach.

Indeed, the computational performance of the MFN method comes down to computing the matrix product. As mentioned, as only $\mathbf{Db}$ is required in this method, it is not necessary to explicitly store the matrix $\mathbf{D}$ in memory. This so-called "matrix-free method" was implemented and tested successfully. As a first test, we used a simple implementation that brute-force recalculated the elements of $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ when required and avoided storing these elements in memory. While the memory usage decreased as expected, the time necessary to recompute the covariances made the method uncompetitive with the stored-in-memory matrix approach. The matrix-free implementation took 29:19 minutes on 386 processors for 4.5k observations versus just 5:40 minutes with a stored matrix. A more suitable matrix-free implementation such one based on FFT would make this feature of the matrix function algorithm more attractive. Additional research is required in this area.

As an additional note, the MFN approach for solving the mean $\mathbf{x} = f_1(\mathbf{D})(\mathbf{y} - \overline{H(\mathbf{X}_f)})$ was compared with the more traditional method of solving for $\mathbf{Dx} = \mathbf{y} - \overline{H(\mathbf{X}_f)}$ using GMRES. In this particular case, the MFN was found to be competitive with GMRES. This may be due to the fact that $\mathbf{D}$ is relatively dense and an efficient pre-conditioner for use with GMRES was not found. Regardless, the novel contribution here is computing the more difficult $f_2(\mathbf{D})(\mathbf{0} - H\mathbf{X})$ using MFN.

6. Discussion and conclusions

In this work we describe the utilization of matrix functions, a powerful linear algebra tool, to derive numerically accurate and efficient solutions of the ESRF equations. With this method, high-rank localized covariance matrices can be applied consistently in such a way that the final analysis does not depend upon the ordering of observations. For the number of observations investigated, this method is roughly competitive in terms of wall-time with the highly efficient serial filter of AC07.

The matrix function approach is built on the Arnoldi iteration, which provides a basis for the Krylov subspace spanned by the covariance matrix of the forward-computed observations $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ and a vector $\mathbf{b}$. This basis allows for evaluation of the ESRF matrix functions over a much smaller, upper Hessenberg matrix. The Scalable Library for Eigenproblem Computation (SLEPc, Hernandez et al. 2005) includes an efficient implementation of the matrix function method along with the Eiermann-Ernst restart (Eiermann and Ernst 2006). Only the matrix-vector product is required, which can be used to provide matrix-free implementations, although for performance reasons storing the entire sparse $\mathbf{C}_{H\mathbf{x},H\mathbf{x}}$ matrix across processing elements may be preferable as shown in our case. The ability to consistently incorporate high-rank covariance models with a known error bounds provides a platform to investigate hybrid ensemble/climatological covariances as well as observation versus model space covariance issues.

Additional effort will be needed to fully understand the computational performance of this method in comparison to other existing parallel EnKF techniques, but a few basic conclusions can be drawn. First, in comparison to the S17 eigenpair solution method of Steward et al. (2017), the matrix function approach scales much better as a function of the number of observations assimilated and uses less memory while maintaining independence of observation ordering and achieving nearly identical numerical results. Second, while this method and the Consistent Hybrid Ensemble Filter of Bishop et al. (2015, CHEF) are similarly independent of the order of observations for high-rank covariance models, as the matrix function approach applies the high-rank covariance matrices globally, it may be more computationally efficient than CHEF (which applies the matrices locally), especially for long localization lengths. This approach also solves the ESRF equations rather using than perturbed observations. Finally, the matrix function method is competitive with the serial AC07 implementation of Anderson and Collins (2007) in terms of wall-time for the cases tested here. While it uses more memory, the matrix function approach is shown to be more faithful to the eigenpair-based solution of the ESRF equations than AC07. It is unknown if this additional precision will have a positive impact on forecasts. The recent work of Emanuel and Zhang (2017) demonstrates the crucial impact of inner core moisture on TC predictability, and the two serial AC07 analyses shown in fig. 2 with merely different observation orderings differ on the extent of dry air near the inner core. As shown, the two water vapor analyses for this difficult first-cycle TC case can differ by up to 3 g kg^{-1}, and therefore it is reasonable to expect the two serial analyses shown in figure 2 may produce qualitatively different medium-term forecasts. A method that can increase fidelity to the ESRF equations, known to be the minimum variance solution (e.g. Bishop et al. 2015), for tropical cyclone cases may be worth the additional computational expense. Due to the efficiency and ease of implementation of the serial filter, continued research into minimizing observation ordering impact is also likely to be beneficial.

Comparison of this method to other local analysis methods remains more unclear. The performance of local analysis methods is most critically related to the radius of influence. For large radii as considered here, this would likely make local analysis methods inefficient as the problem for each local grid point becomes nearly as large as the entire domain. However, in such cases, when sample-based covariance localization is utilized with the ESRF approach, the matrix function approach could also potentially be used to improve performance versus $O(n^3)$ algorithms such as finding eigenpairs or the Cholesky decomposition. This may be unnecessary, however, if the number of local observations does not exceed $\approx 10^2$.

At the moment, a major weakness of the non-local matrix function approach in comparison to the AC07 serial approach is the memory usage scaling. Extrapolating the results presented in figure 6 on 386 processors and keeping the number of processors constant, with approximately 80k observations (assuming quadratic growth) to 115k observations (assuming linear growth) the matrix function approach would run out of memory. By comparison, the serial filter would run out of memory (assuming linear growth) at approximately 3.2 million observations. A matrix free implementation would address this issue. Since in the matrix function approach, computational performance comes down efficient methods of applying the matrix product, we aim to investigate application of the modulation product of Bishop and Hodyss (2009) to apply correlations in order improve the memory scaling issue. In the meantime, batch processing of large numbers of observations is one potential work-around.

The algorithm described in this paper requires a distributed sparse matrix implementation such as that available in the Portable Extensible Toolkit for Scientific Computing (PETSc, Balay et al.

1997, 2016, 2017) which SLEPc is built upon. In addition, the restarted Arnoldi process (including a numerically stable parallel Gram-Schmidt orthogonalization process) must be implemented to estimate the required reduced-order matrix function products. When using the SLEPc library that provides this functionality, this approach is not more difficult than the eigenpair implementation of S17. However, either implementation is certainly more complex than the serial approximation.

Finally, while the order-dependency issue shown here is non-trivial, the TC first-cycle case is likely to be a "worst-case" scenario due to the highly non-linear nature of feature misalignment. While Nerger (2015) hypothesized that the effect of the observation-order dependency in the serial implementation is small when the analysis is not far from the prior, the filter described here may be useful to test the practical effect of this hypothesis in a variety of large-scale cases and to develop mitigation solutions for the serial approach when necessary.

Acknowledgments. This research was partially funded by the NOAA Hurricane Forecast Improvement Project award number NA14NWS4680022. This work was partially supported by Agencia Estatal de Investigación (AEI) under grant TIN2016-75985-P, which includes European Commission ERDF funds. Alejandro Lamas Daviña was supported by the Spanish Ministry of Education, Culture and Sport through a grant with reference FPU13-06655. The fourth author's work was in part carried out under the auspices of CIMAS, a joint institute of the University of Miami and NOAA, cooperative agreement NA15OAR4320064. The authors acknowledge the NOAA Research and Development High Performance Computing Program for providing computing and storage resources that have contributed to the research results reported within this paper. URL: `http://rdhpcs.noaa.gov`. We thank Jeff Anderson, Shu-Chih Yang, and three anonymous reviewers for their helpful comments and contributions. We also thank Hui Christophersen for providing technical assistance.

References

Aberson, S. D., A. Aksoy, K. J. Sellwood, T. Vukicevic, and X. Zhang, 2015: Assimilation of High-Resolution Tropical Cyclone Observations with an Ensemble Kalman Filter Using HEDAS: Evaluation of 2008–11 HWRF Forecasts. *Monthly Weather Review*, **143 (2)**, 511–523, doi:10.1175/MWR-D-14-00138.1.

Afanasjew, M., M. Eiermann, O. G. Ernst, and S. Güttel, 2008: Implementation of a restarted Krylov subspace method for the evaluation of matrix functions. *Linear Algebra and its Applications*, **429 (10)**, 2293–2314, doi:10.1016/j.laa.2008.06.029.

Aksoy, A., 2013: Storm-Relative Observations in Tropical Cyclone Data Assimilation with an Ensemble Kalman Filter. *Monthly Weather Review*, **141 (2)**, 506–522, doi:10.1175/MWR-D-12-00094.1.

Aksoy, A., S. D. Aberson, T. Vukicevic, K. J. Sellwood, S. Lorsolo, and X. Zhang, 2013: Assimilation of High-Resolution Tropical Cyclone Observations with an Ensemble Kalman Filter Using NOAA/AOML/HRD's HEDAS: Evaluation of the 2008–11 Vortex-Scale Analyses. *Monthly Weather Review*, **141 (6)**, 1842–1865, doi:10.1175/MWR-D-12-00194.1.

Aksoy, A., S. Lorsolo, T. Vukicevic, K. J. Sellwood, S. D. Aberson, and F. Zhang, 2012: The HWRF Hurricane Ensemble Data Assimilation System (HEDAS) for High-Resolution Data: The Impact of Airborne Doppler Radar Observations in an OSSE. *Monthly Weather Review*, **140 (6)**, 1843–1862.

Anderson, J. L., 2001: An Ensemble Adjustment Kalman Filter for Data Assimilation. *Monthly Weather Review*, **129**, 2884–2903, doi:10.1175/1520-0493(2001)129⟨2884:AEAKFF⟩2.0.CO;2.

Anderson, J. L., 2003: A Local Least Squares Framework for Ensemble Filtering. *Monthly Weather Review*, **131 (4)**, 634–642, doi:10.1175/1520-0493(2003)131⟨0634:ALLSFF⟩2.0.CO;2.

Anderson, J. L., 2012: Localization and Sampling Error Correction in Ensemble Kalman Filter Data Assimilation. *Monthly Weather Review*, **140 (7)**, 2359–2371, doi:10.1175/MWR-D-11-00013.1.

Anderson, J. L., and N. Collins, 2007: Scalable Implementations of Ensemble Filter Algorithms for Data Assimilation. *Journal of Atmospheric and Oceanic Technology*, **24 (8)**, 1452–1463, doi:10.1175/JTECH2049.1.

Arnoldi, W. E., 1951: The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, **9 (1)**, 17–29, doi:10.1090/qam/42792.

Balay, S., W. D. Gropp, L. C. McInnes, and B. F. Smith, 1997: Efficient management of parallelism in object-oriented numerical software libraries. *Modern software tools for scientific computing*, Springer, 163–202.

Balay, S., and Coauthors, 2016: PETSc Users Manual. Tech. Rep. ANL-95/11 - Revision 3.7, Technical report, Argonne National Laboratory (ANL). URL http://www.mcs.anl.gov/petsc/petsc-current/docs/manual.pdf.

Balay, S., and Coauthors, 2017: PETSc: Home Page. URL https://www.mcs.anl.gov/petsc/.

Bessho, K., and Coauthors, 2016: An Introduction to Himawari-8/9 — Japan's New-Generation Geostationary Meteorological Satellites. *Journal of the Meteorological Society of Japan. Ser. II*, **94 (2)**, 151–183, doi:10.2151/jmsj.2016-009.

Bishop, C. H., B. J. Etherton, and S. J. Majumdar, 2001: Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly Weather Review*, **129 (3)**, 420–436.

Bishop, C. H., and D. Hodyss, 2009: Ensemble covariances adaptively localized with ECO-RAP. Part 2: a strategy for the atmosphere. *Tellus A*, **61 (1)**, 97–111, doi:10.1111/j.1600-0870.2008.00372.x.

Bishop, C. H., B. Huang, and X. Wang, 2015: A Nonvariational Consistent Hybrid Ensemble Filter. *Monthly Weather Review*, **143 (12)**, 5073–5090, doi:10.1175/MWR-D-14-00391.1.

Björck, Å., 1994: Numerics of Gram-Schmidt orthogonalization. *Linear Algebra and its Applications*, **197-198 (Supplement C)**, 297–316, doi:10.1016/0024-3795(94)90493-6.

Burgers, G., P. J. van Leeuwen, and G. Evensen, 1998: Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, **126 (6)**, 1719–1724.

Campbell, W. F., C. H. Bishop, and D. Hodyss, 2010: Vertical Covariance Localization for Satellite Radiances in Ensemble Kalman Filters. *Monthly Weather Review*, **138 (1)**, 282–290, doi:10.1175/2009MWR3017.1.

Chang, C.-C., S.-C. Yang, and C. Keppenne, 2014: Applications of the Mean Recentering Scheme to Improve Typhoon Track Prediction: A Case Study of Typhoon Nanmadol (2011). *Journal of the Meteorological Society of Japan. Ser. II*, **92 (6)**, 559–584, doi:10.2151/jmsj.2014-604.

Christophersen, H., A. Aksoy, J. Dunion, and K. Sellwood, 2017: The Impact of NASA Global Hawk Unmanned Aircraft Dropwindsonde Observations on Tropical Cyclone Track, Intensity, and Structure: Case Studies. *Monthly Weather Review*, **145 (5)**, 1817–1830, doi:10.1175/MWR-D-16-0332.1.

Deadman, E., N. J. Higham, and R. Ralha, 2012: Blocked Schur Algorithms for Computing the Matrix Square Root. *Applied Parallel and Scientific Computing*, Springer, Berlin, Heidelberg, 171–182, Lecture Notes in Computer Science, doi:10.1007/978-3-642-36803-5_12.

Eiermann, M., and O. Ernst, 2006: A Restarted Krylov Subspace Method for the Evaluation of Matrix Functions. *SIAM Journal on Numerical Analysis*, **44 (6)**, 2481–2504, doi:10.1137/050633846.

Eiermann, M., O. Ernst, and S. Güttel, 2011: Deflated Restarting for Matrix Functions. *SIAM Journal on Matrix Analysis and Applications*, **32 (2)**, 621–641, doi:10.1137/090774665.

Emanuel, K., and F. Zhang, 2017: The Role of Inner-Core Moisture in Tropical Cyclone Predictability and Practical Forecast Skill. *Journal of the Atmospheric Sciences*, **74 (7)**, 2315–2324, doi:10.1175/JAS-D-17-0008.1.

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, **99 (10)**, 143–10, doi:/doi/10.1029/94JC00572.

Frayssé, V., L. Giraud, and H. Kharraz-Aroussi, 1998: On the influence of the orthogonalization scheme on the parallel performance of GMRES. *Euro-Par'98 Parallel Processing*, Springer, Berlin, Heidelberg, 751–762, Lecture Notes in Computer Science, doi:10.1007/BFb0057927.

Frommer, A., K. Lund, M. Schweitzer, and D. Szyld, 2017: The Radau–Lanczos Method for Matrix Functions. *SIAM Journal on Matrix Analysis and Applications*, 710–732, doi:10.1137/16M1072565.

Gaspari, G., and S. E. Cohn, 1999: Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, **125 (554)**, 723–757, doi: 10.1002/qj.49712555417.

Godinez, H. C., and J. D. Moulton, 2012: An efficient matrix-free algorithm for the ensemble Kalman filter. *Computational Geosciences*, **16 (3)**, 565–575, doi:10.1007/s10596-011-9268-9.

Golub, G. H., and C. F. Van Loan, 1996: *Matrix computations*, Vol. 3. Johns Hopkins University Press.

Gopalakrishnan, S. G., F. Marks, X. Zhang, J.-W. Bao, K.-S. Yeh, and R. Atlas, 2010: The Experimental HWRF System: A Study on the Influence of Horizontal Resolution on the Structure and Intensity Changes in Tropical Cyclones Using an Idealized Framework. *Monthly Weather Review*, **139 (6)**, 1762–1784, doi:10.1175/2010MWR3535.1.

Hamill, T. M., J. S. Whitaker, and C. Snyder, 2001: Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter. *Monthly Weather Review*, **129 (11)**, 2776–2790, doi:10.1175/1520-0493(2001)129⟨2776:DDFOBE⟩2.0.CO;2.

Hernandez, V., J. E. Roman, and A. Tomas, 2007: Parallel Arnoldi eigensolvers with enhanced scalability via global communications rearrangement. *Parallel Computing*, **33 (7)**, 521–540, doi:10.1016/j.parco.2007.04.004.

Hernandez, V., J. E. Roman, and V. Vidal, 2005: SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems. *ACM Transactions on Mathematical Software*, **31 (3)**, 351–362, doi:10.1145/1089014.1089019.

Higham, N., 2008: *Functions of Matrices*. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics.

Higham, N. J., 1987: Computing real square roots of a real matrix. *Linear Algebra and its Applications*, **88**, 405–430, doi:10.1016/0024-3795(87)90118-2.

Hochbruck, M., and C. Lubich, 1997: On Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM Journal on Numerical Analysis*, **34 (5)**, 1911–1925, doi:10.1137/S0036142995280572.

Houtekamer, P. L., B. He, and H. L. Mitchell, 2013: Parallel Implementation of an Ensemble Kalman Filter. *Monthly Weather Review*, **142 (3)**, 1163–1182, doi:10.1175/MWR-D-13-00011.1.

Houtekamer, P. L., and H. L. Mitchell, 1998: Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, **126 (3)**, 796–811.

Houtekamer, P. L., and H. L. Mitchell, 2001: A Sequential Ensemble Kalman Filter for Atmospheric Data Assimilation. *Monthly Weather Review*, **129 (1)**, 123–137, doi:10.1175/1520-0493(2001)129⟨0123:ASEKFF⟩2.0.CO;2.

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

Kalman, R. E., 1960: A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, **82 (1)**, 35–45.

Keppenne, C. L., and M. M. Rienecker, 2002: Initial Testing of a Massively Parallel Ensemble Kalman Filter with the Poseidon Isopycnal Ocean General Circulation Model. *Monthly Weather Review*, **130 (12)**, 2951–2965, doi:10.1175/1520-0493(2002)130⟨2951:ITOAMP⟩2.0.CO;2.

Kotsuki, S., S. J. Greybush, and T. Miyoshi, 2017: Can We Optimize the Assimilation Order in the Serial Ensemble Kalman Filter? A Study with the Lorenz-96 Model. *Monthly Weather Review*, **145 (12)**, 4977–4995, doi:10.1175/MWR-D-17-0094.1.

Miyoshi, T., and Coauthors, 2016: “Big Data Assimilation” – Revolutionizing Severe Weather Prediction. *Bulletin of the American Meteorological Society*, **97 (8)**, 1347–1354, doi:10.1175/BAMS-D-15-00144.1.

Nerger, L., 2015: On Serial Observation Processing in Localized Ensemble Kalman Filters. *Monthly Weather Review*, **143 (5)**, 1554–1567, doi:10.1175/MWR-D-14-00182.1.

Nino-Ruiz, E. D., A. Sandu, and J. Anderson, 2015: An efficient implementation of the ensemble Kalman filter based on an iterative Sherman–Morrison formula. *Statistics and Computing*, **25 (3)**, 561–577, doi:10.1007/s11222-014-9454-4.

Nino-Ruiz, E. D., A. Sandu, and X. Deng, 2017: A parallel implementation of the ensemble Kalman filter based on modified Cholesky decomposition. *Journal of Computational Science*, doi:10.1016/j.jocs.2017.04.005.

Ott, E., and Coauthors, 2002: A Local Ensemble Kalman Filter for Atmospheric Data Assimilation. *arXiv:physics/0203058*, arXiv: physics/0203058.

Rogers, R. F., J. A. Zhang, J. Zawislak, H. Jiang, G. R. Alvey, E. J. Zipser, and S. N. Stevenson, 2016: Observations of the Structure and Evolution of Hurricane Edouard (2014) during Intensity Change. Part II: Kinematic Structure and the Distribution of Deep Convection. *Monthly Weather Review*, **144 (9)**, 3355–3376, doi:10.1175/MWR-D-16-0017.1.

Saad, Y., 1992: Analysis of Some Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM Journal on Numerical Analysis*, **29 (1)**, 209–228, doi:10.1137/0729014.

Sakov, P., and L. Bertino, 2011: Relation between two common localisation methods for the EnKF. *Computational Geosciences*, **15 (2)**, 225–237, doi:10.1007/s10596-010-9202-6.

Schmit, T. J., P. Griffith, M. M. Gunshor, J. M. Daniels, S. J. Goodman, and W. J. Lebair, 2016: A Closer Look at the ABI on the GOES-R Series. *Bulletin of the American Meteorological Society*, **98 (4)**, 681–698, doi:10.1175/BAMS-D-15-00230.1.

Steward, J. L., A. Aksoy, and Z. S. Haddad, 2017: Parallel Direct Solution of the Ensemble Square Root Kalman Filter Equations with Observation Principal Components. *Journal of Atmospheric and Oceanic Technology*, **34 (9)**, 1867–1884, doi:10.1175/JTECH-D-16-0140.1.

Tippett, M. K., J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker, 2003: Ensemble Square Root Filters. *Monthly Weather Review*, **131 (7)**, 1485–1490, doi:10.1175/1520-0493(2003)131⟨1485:ESRF⟩2.0.CO;2.

Van Der Vorst, H. A., 1987: An iterative solution method for solving f(A)x = b, using Krylov subspace information obtained for the symmetric positive definite matrix A. *Journal of Computational and Applied Mathematics*, **18 (2)**, 249–263, doi:10.1016/0377-0427(87)90020-3.

Vukicevic, T., A. Aksoy, P. Reasor, S. D. Aberson, K. J. Sellwood, and F. Marks, 2013: Joint Impact of Forecast Tendency and State Error Biases in Ensemble Kalman Filter Data Assimilation of Inner-Core Tropical Cyclone Observations. *Monthly Weather Review*, **141 (9)**, 2992–3006, doi:10.1175/MWR-D-12-00211.1.

Wang, Y., Y. Jung, T. A. Supinie, and M. Xue, 2013: A Hybrid MPI–OpenMP Parallel Algorithm and Performance Analysis for an Ensemble Square Root Filter Designed for Multi-scale Observations. *Journal of Atmospheric and Oceanic Technology*, **30 (7)**, 1382–1397, doi:10.1175/JTECH-D-12-00165.1.

Whitaker, J. S., and T. M. Hamill, 2002: Ensemble Data Assimilation without Perturbed Observations. *Monthly Weather Review*, **130 (7)**, 1913–1924, doi:10.1175/1520-0493(2002)130⟨1913:EDAWPO⟩2.0.CO;2.

Zawislak, J., H. Jiang, G. R. Alvey, E. J. Zipser, R. F. Rogers, J. A. Zhang, and S. N. Stevenson, 2016: Observations of the Structure and Evolution of Hurricane Edouard (2014) during Intensity Change. Part I: Relationship between the Thermodynamic Structure and Precipitation. *Monthly Weather Review*, **144 (9)**, 3333–3354, doi:10.1175/MWR-D-16-0018.1.

Zhang, S., M. J. Harrison, A. T. Wittenberg, A. Rosati, J. L. Anderson, and V. Balaji, 2005: Initialization of an ENSO Forecast System Using a Parallelized Ensemble Filter. *Monthly Weather Review*, **133 (11)**, 3176–3201, doi:10.1175/MWR3024.1.

LIST OF TABLES

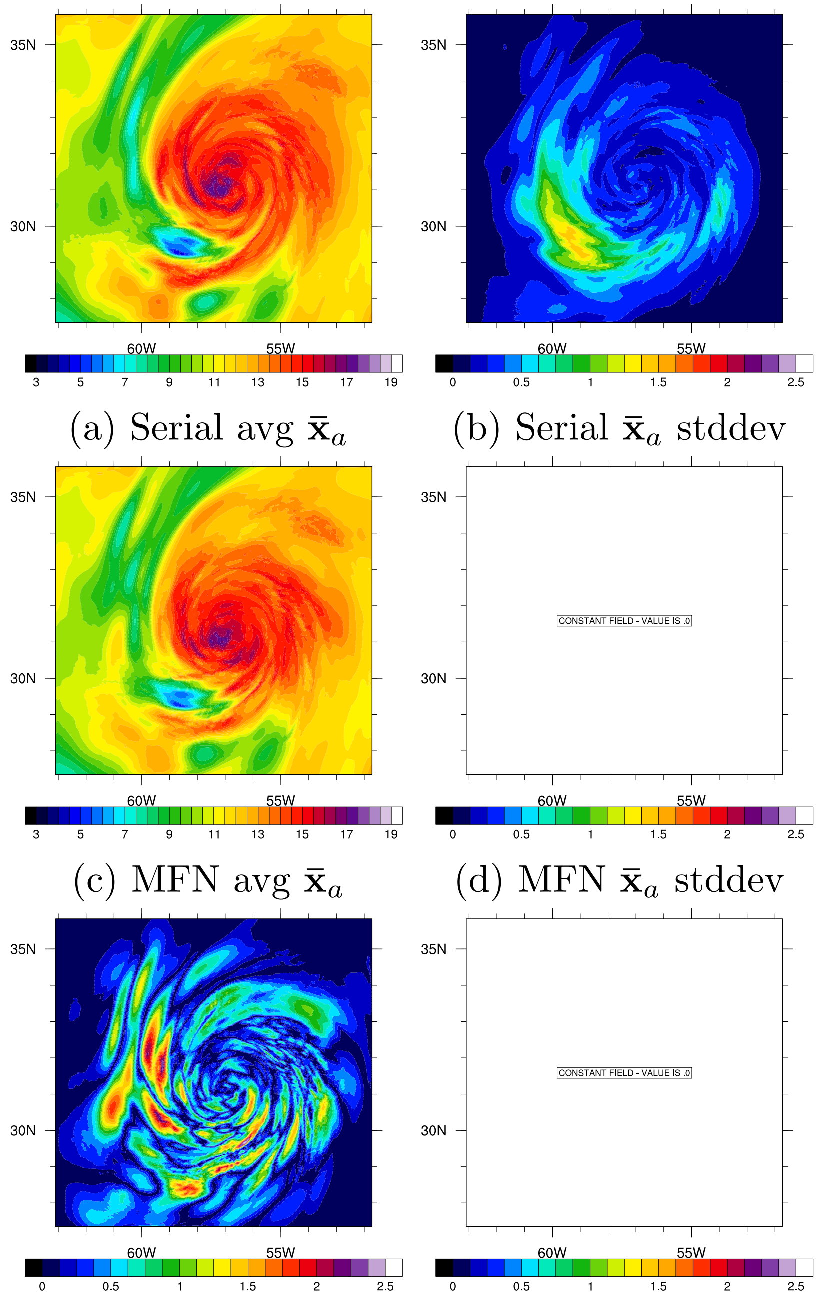
TABLE 1. Time to complete the solution, number of restarts (per control vector), and total number of matrix product evaluations as a function of m, the size of the Krylov subspace before restarting, required to solve the perturbation update matrix function f_2 in equation (24) with $L = 240$ (in equations (5) and (8)) and 17.7K observations as described in section 5. The timings are with a single MPI process on an Intel Core i7 server. Note these times are for a single ensemble member.

m	Time (s)	Restarts	Total evals
25	2.5743e+04	74	1875
50	4.8219e+03	12.2	660
75	3.2298e+03	5	450
100	2.8686e+03	3	400
125	2.6897e+03	2	375
150	3.2460e+03	2	450
175	2.5257e+03	1	350
200	2.8950e+03	1	400

TABLE 2. As in table 1 but with $L = 60$. The reduction in time versus $L = 240$ is due to the increased sparsity of the localization matrices $\rho_{y,y}$ and $\rho_{x,y}$.

m	Time (s)	Restarts	Total evals
25	1.6822e+04	66.2	1705
50	3.8763e+03	15.0333	802
75	2.7369e+03	6.9	593
100	2.2974e+03	4	500
125	2.2872e+03	3	500
150	2.0749e+03	2	450
175	2.4319e+03	2	525
200	2.2491e+03	1.43333	487

LIST OF FIGURES



(a) Serial avg $\bar{\mathbf{x}}_a$ (b) Serial $\bar{\mathbf{x}}_a$ stddev

(c) MFN avg $\bar{\mathbf{x}}_a$ (d) MFN $\bar{\mathbf{x}}_a$ stddev

(e) |Avg serial - EPS| (f) |Avg MFN - EPS|

FIG. 1. Comparison between water vapor (g kg^{-1}) at level 20 (of 60 total, corresponding to a height of approximately 2.5 km) of the Hurricane Edouard single cycle case of Christophersen et al. (2017) with 15.2K observations and $L = 240$ as described in S17. Ten different random orderings of observations were used. (a) The average of the ten AC07 $\bar{\mathbf{x}}_a$ analyses. (b) The standard deviation of these ten AC07 $\bar{\mathbf{x}}_a$ analyses. (c) The average of the ten matrix function (MFN) $\bar{\mathbf{x}}_a$ analyses. (d) The standard deviation of the MFN analyses, which is less than 10^{-7} at all points. (e) The absolute difference between (a) and the EPS solution. (f) The absolute difference between (c) and the EPS solution (also less than 10^{-7} for all points).

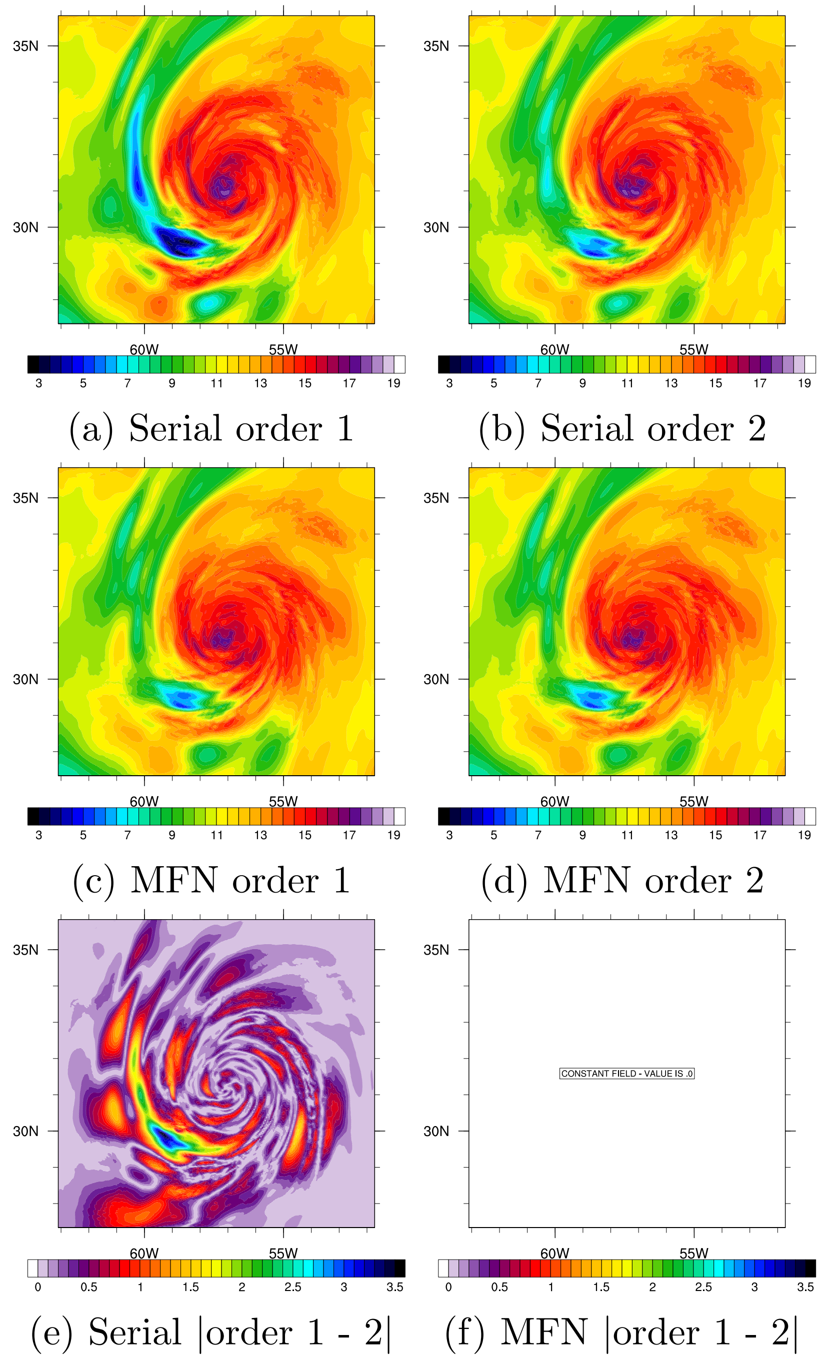


FIG. 2. Comparison between analyzed $\bar{\mathbf{x}}_a$ level 20 water vapor as in fig. 1 for two of the ten different random orderings of observations. (a) The serial filter of Anderson and Collins (2007) with ordering 1 and (b) ordering 2. (c) MFN analyzed $\bar{\mathbf{x}}_a$ ordering 1. (d) MFN $\bar{\mathbf{x}}_a$ ordering 2. (e) The absolute value difference between (a) and (b). (f) The difference between the two MFN orderings in (c) and (d), which is less than 10^{-7}. The difference between the MFN and EPS analysis for this case is also less than 10^{-7} at all levels.

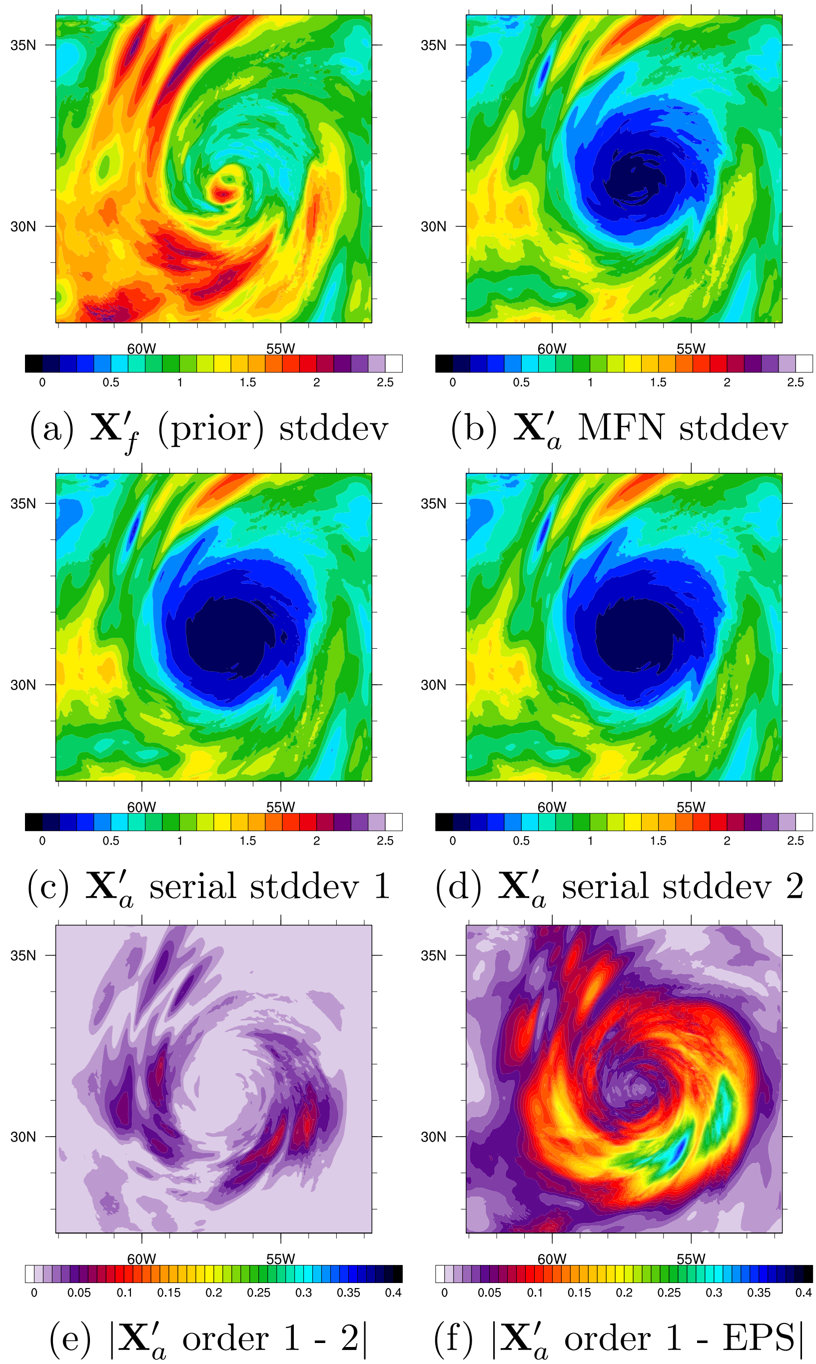
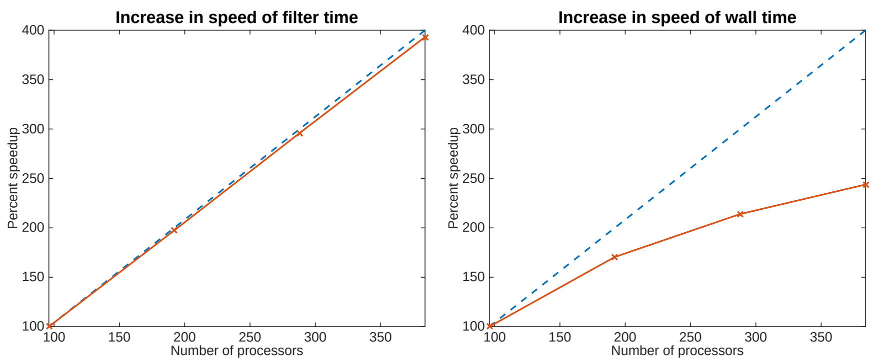


FIG. 3. Ensemble spread (i.e. standard deviations over the ensemble) of water vapor (g kg^{-1}) at level 20 as in figures 1 and 2. Here the first two random orderings of observations were used as in fig. 2. (a) The standard deviation of the prior distribution $\mathbf{X}'_f$ at this level. (b) The standard deviation of the MFN posterior distribution $\mathbf{X}'_a$ (ordering 1, 2, and the EPS solution are the same to with 10^{-7}). (c) Standard deviation of $\mathbf{X}'_a$ for AC07 ordering 1 and (d) ordering 2. (e) The absolute difference between the serial analysis with ordering 1 and 2 from (c) and (d). (f) The absolute difference between the EPS solution and ordering 1 from (c).



(a) Filter speed-up

(b) Total speed-up

FIG. 4. (a) Speed increase of applying the MFN filter, including the time to calculate $(\mathbf{E}|\mathbf{g})$ using the matrix function approach and solve equation (1) as a function of number of processing elements with the number of observations fixed at 17k. The speedup is nearly linear and is dominated by the time applying $\mathbf{C}_{\mathbf{x},H\mathbf{x}}$. This should be compared with figure 6d) from S17 which likewise shows a nearly linear speed increase as a function of number of processors during filter time. (b) Total speed increase of wall-time including disk reads and writes. As the process is I/O bound, the total speed increase is sublinear. Compare with figure 6f) from S17 which likewise shows a sub-linear increase (and even an eventual decrease) as a function of total wall time due to degradation in parallel I/O performance.

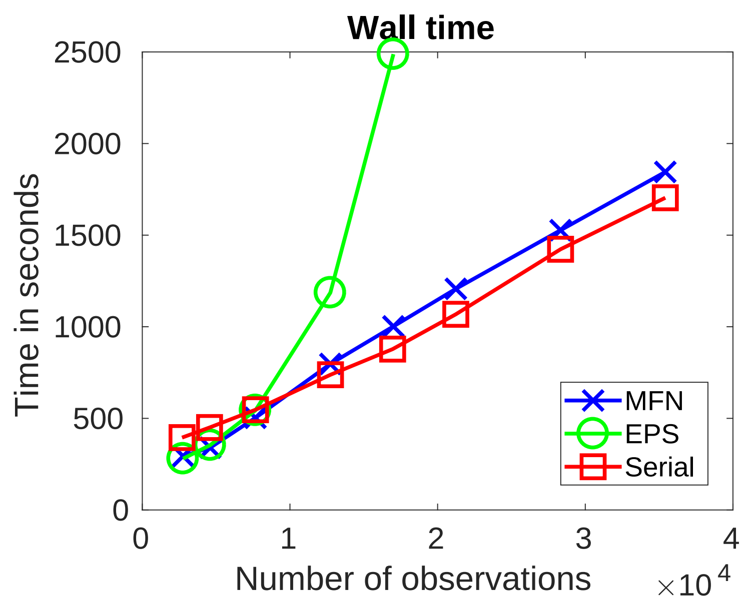


FIG. 5. Scaling as a function of number of observations with 386 processors. The MFN approach described in this paper appears to scale approximately linearly ($y = 4.86 \times 10^{-2}x + 152$), while the EPS scales consistent with a cubic fit ($y = 4.43 \times 10^{-10}x^3 + 302$). The serial filter of Anderson and Collins (2007) likewise scales linearly ($y = 4.04 \times 10^{-2}x + 224.72$). Times longer than 2500 seconds for the EPS solution are not shown.

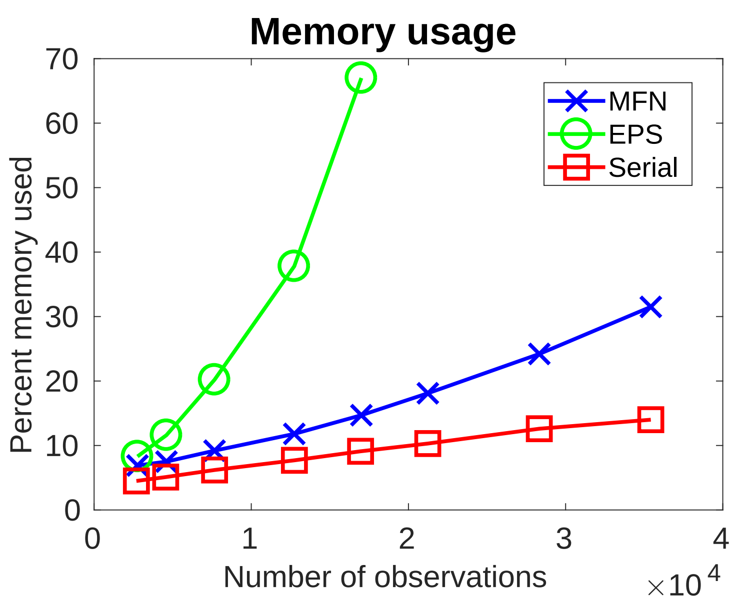


FIG. 6. Memory usage as a function of the number of observations with 386 processors. The EPS scales cubically as predicted by theory, while the serial filter of Anderson and Collins (2007) scales linearly in memory usage. The MFN approach apparently scales worse than linearly. A matrix-free implementation of MFN improves memory scaling.