# Sparsity-Based Kalman Filters for Data Assimilation

**Wei Kang and Liang Xu**

**Abstract** Several variations of the Kalman filter, such as the extended Kalman filter (EKF) and the unscented Kalman filter (UKF), are widely used in science and engineering applications. However, traditional UKFs or EKFs cannot assimilate big data sets associated with models that have high dimensions, such as those in operational numerical weather prediction. In this chapter, we introduce two sparsity-based Kalman filters, namely the sparse-UKF and the progressive-EKF. The filters are designed specifically for problems with high dimensions. Different from ensemble Kalman filters (EnKFs) in which the error covariance is approximated using a set of dense ensemble vectors, the algorithms developed in this chapter are based on the sparse matrix approximation of error covariance. The new algorithms enjoy several advantages. The error covariance has full rank without being limited within a subspace generated by a set of ensembles. In addition to the estimated states, the algorithms provide updated error covariance in every assimilation cycle. Taking the advantage of sparsity, the required memory size and computational load can be significantly reduced.

## 1 Introduction

For dynamical systems, data assimilation is a process that integrates observational data with a numerical model for the purpose of estimating the system's state. Data assimilation is essential to numerical weather prediction (NWP). The estimate of the state value is used as the initial condition for weather forecast. If the dimension is relatively low and the data set is small, various linear and nonlinear estimators can

This work was supported in part by U.S. Naval Research Laboratory—Monterey, CA.

W. Kang (✉)
Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA, USA
e-mail: wkang@nps.edu

L. Xu
Naval Research Laboratory, Monterey, CA, USA
e-mail: liang.xu@nrlmry.navy.mil

S. K. Park and L. Xu (eds.), *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications (Vol. IV)*, https://doi.org/10.1007/978-3-030-77722-7_4

be found in the literature that have optimal or suboptimal performances. However, to assimilate big data sets with models that have high dimensions, such as those in operational NWP systems with tens of millions of variables, achieving reliable state estimation and error probability distributions is a challenging problem that have been studied for decades with a huge literature.

There are two categories of methods widely used in NWP, namely the variational method and the ensemble Kalman filter (EnKF) (Xu et al. 2005; Houtekamer and Zhang 2016). The former is based on a weighted least-square optimization, such as the four dimensional variational data assimilation (4D-Var) in a fixed time window or the three dimensional version (3D-Var) that excludes the time variable. The EnKF algorithm is based on the Kalman filter except that the error covariance is approximated using a set of ensembles. 4D-Var methods are used in operational NWP systems by many meteorological centers. While it serves as an effective method of data assimilation, 4D-Var algorithms have difficulty to explicitly track the evolution of error covariance within its estimation process due to high computational costs and input/output (I/O) loads required by the process of high dimensional matrices. EnKF, on the other hand, updates information about the error covariance in the form of ensembles. However, it is common in practical applications that the number of vectors in an ensemble is significantly smaller than the number of state variables. As a result, the rank deficiency of error covariance tends to deteriorate the integrity of the estimation process unless remedies to the algorithm, such as localization and covariance inflation, are applied.

Different types of Kalman filters have been developed and widely used in science and engineering applications, such as the EnKF, the extended Kalman filter (EKF) and the unscented Kalman filter (UKF). In this chapter, we introduce two sparsity-based Kalman filters, namely the sparse-UKF and the progressive-EKF. The goal of the work is to explore innovative ideas that take the advantage of the sparsity structure of matrices so that analysis and error covariance can be updated effectively and efficiently without the drawback of rank deficiency. The filters are developed specifically for problems with high dimensions. Different from EnKFs in which the error covariance is represented by a set of dense vectors in an ensemble, the new algorithms in this chapter are based on a sparse but full rank matrix as an approximation of the error covariance. This is made possible because of two assumptions: (a) the error covariance is approximately a sparse matrix; (b) the system model is component based, i.e. the state vectors are divided into components that can be computed independently in parallel. In Sect. 2, analysis is provided to justify that assumption (a) is expected to hold for a large family of system models. Assumption (b) is about the numerical method used for the system model, a topic that is not addressed in this chapter. In Sect. 3, the sparse-UKF is introduced. Its performance is exemplified using a Lorenz-96 model. In Sect. 4, the progressive-EKF is introduced and exemplified using the same Loren-96 model as in Sect. 3. In all examples, the new estimation methods developed in this chapter are compared to an EnKF as well as a traditional UKF with a full rank dense covariance.

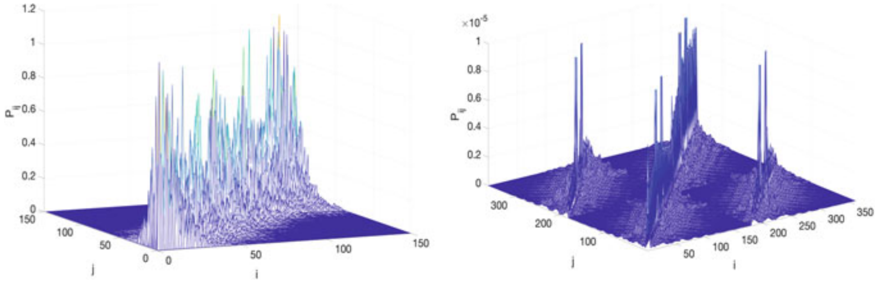


**Fig. 1** Examples of Kalman filter error covariance

## 2 The Sparsity of Error Covariance

Shown in Fig. 1 are two error covariance matrices of Kalman filters, one for a linear system of ordinary differential equations and the other for a discretized shallow water equation. The dimensions of the state spaces are $n = 150$ and $n = 350$, respectively. The $x$- and $y$-axes represent row and column indices, $i$ and $j$; the $z$-axis represents the absolute value of the error covariance, $|P_{ij}|$. Both matrices are approximately sparse, i.e., the majority of entries are relatively small. The matrices have peak value around one or multiple diagonals only. Approximating the covariance using a sparse matrix by setting all small entries to zeros, one can significantly reduce the computational cost, I/O loads and the amount of memory usage.

The approximate sparsity shown in Fig. 1 is not unusual. In fact, a theorem in Kang and Xu (2021) indicates that this type of sparse covariance is expected for a family of dynamic systems. In the following, a matrix $A$ is said to be banded with bandwidth $s \geq 0$ if $A_{ij} = 0$ whenever $|i - j| > s$. If $s = 0$, then the matrix is diagonal. We say that a symmetric matrix $P$ is less than or equal to another symmetric matrix $G$, or $P \leq G$, if $G - P$ is positive semidefinite. Consider the following system of ordinary differential equations (ODEs)

$$\begin{aligned} \dot{x}(t) &= Ax(t) + \eta(t), \quad && x, \eta \in \mathbb{R}^n, \\ y(t) &= Hx(t) + \delta(t), \quad && y, \delta \in \mathbb{R}^m \end{aligned} \tag{1}$$

where $x$ is the state variable, $\dot{x}$ represents its time derivative, $y$ is the observation variable, $\eta$ and $\delta$ are zero-mean Gaussian white noise with covariances $Q$ and $R$, respectively.

**Theorem** (Kang and Xu 2021) *Suppose that $A$ and $Q$ in (1) are banded. Let $P(t)$ be the error covariance of the Kalman filter that estimates $x(t)$. Then*

$$0 \leq P(t) \leq e^{At} P(0) e^{A^T t} + G^C(t) \tag{2}$$

*where $G^C(t)$ is a symmetric matrix. Its entries have an upper bound*

$$|(G^C(t))_{ij}| \leq \frac{\bar{G}\alpha^{(|i-j|+\beta)/\gamma}}{((|i-j|+\beta)/\gamma)^{(|i-j|+\beta)/\gamma}}$$

*for some constants $\bar{G}$, $\alpha$, $\beta$, and $\gamma$. The upper bound decreases at a rate greater than exponential as $|i-j| \to \infty$.*

This result implies that $|(G^C(t))_{ij}|$ is almost zero when $|i-j|$ is large. Although $|i-j|$ is bounded by $n-1$, the result is applicable if $n$ is significantly larger than the bandwidth of the banded matrices $A$ and $Q$. In this case, $G^C$ is approximately a sparse matrix. For the term of initial error covariance, $e^{At}P(0)e^{A^Tt}$, it can be proved that this matrix is also approximately sparse if $P(0)$ is banded with a bandwidth significantly smaller than $n$. If the ODE in (1) is the discretization of partial differential equations, which is the case in many NWP problems, then $A$ is banded provided that local discretization algorithms are used. Inspired by this theorem, we assume in the following sections that the covariance of Kalman filters can be approximated by a sparse matrix. We present two new algorithms of data assimilation in which the matrix of error covariance is computationally tractable.

## 3 Sparse-UKF

Consider a dynamical system model in which the state variable is $x(t)$, where $t = 1, 2, 3, \ldots$ represents time steps. The value of observation at $t = k$ is denoted by $y(k)$. The system model is defined as follows,

$$\begin{aligned} x(k) &= \mathcal{M}(x(k-1)) + \eta_{k-1}, & x(k), \eta_{k-1} &\in \mathbb{R}^n, \\ y(k) &= \mathcal{H}(x(k)) + \delta_k, & y_k, \delta_k &\in \mathbb{R}^m, \end{aligned} \tag{3}$$

where $\eta_{k-1}$ is a random variable representing the model error. Its covariance is $Q$. The observational error, $\delta_k$, has a covariance $R$. In data assimilation, the goal is to estimate the value of $x(k)$ given the observations $y(1), y(2), \ldots, y(k)$ and the model (3). If (3) is linear and if all random variables are Gaussian, then the Kalman filter is an optimal state estimator. For nonlinear systems with non-Gaussian randomness, various types of Kalman filters exist in the literature with successful applications in science and engineering. If a system has a very high dimension, the conventional form of Kalman filter based on a dense error covariance is not applicable. In this section, we introduce an algorithm that is a variation of UKF for problems with approximately sparse matrices of covariance.

## 3.1 Sparse Matrix Algebra

In a sparse matrix, most entries are zeros. For some dense matrices in which most entries are relatively small, we approximate them using sparse matrices. In this chapter, we use an underbar to represent an operator that maps a vector or matrix to a sparse one. For instance, given a vector $x \in \mathbb{R}^n$. Let $N_{sp}$ be an integer representing the size of the sparsity and

$$\mathcal{I} = \{i_1, i_2, \ldots, i_{N_{sp}}\}$$

be an index consisting of a sequence of integers. Then the underbar operator maps $x$ to a vector $\underline{x} \in \mathbb{R}^{N_{sp}}$ in which

$$\underline{x}_k = x_{i_k} \text{ for } k = 1, 2, \ldots, N_{sp}.$$

It is equivalently to say that $\underline{x}$ is obtained from $x$ by removing all $x_i$ if $i \notin \mathcal{I}$. Usually, the removed entries are either zeros or relatively small (in absolute value). We would like to emphasize that, although $\underline{x}$ is formally a vector in $\mathbb{R}^{N_{sp}}$, it is treated as a vector in $\mathbb{R}^n$ by setting the $i$th entry zero if $i \notin \mathcal{I}$. This vector in $\mathbb{R}^n$ is, in general, different from the original vector, $x$, if the latter is a dense vector that is only approximately sparse.

Similarly, we can define the underbar operator for matrices. Given $P \in \mathbb{R}^{n\times n}$. Its columns may have different numbers of nonzero, or relatively large, entries. The largest such number is denoted by $N_{sp}$. Then $\underline{P}$ is a set of vectors associated with index sets

$$\underline{P} = \{\underline{P}_1, \underline{P}_2, \ldots, \underline{P}_n\}, \quad \text{sparsity index set } \mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\} \tag{4}$$

where $\underline{P}_i$ associated with $\mathcal{I}_i$ is the sparse vector approximation of the $i$th column of $P$. In all algebraic derivations, $\underline{P}$ is treated as a matrix in $\mathbb{R}^{n\times n}$ in which all entries are zeros except for those included in $\mathcal{I}$.

In sparsity-based algorithms, a full model evaluation is not always necessary. Using a *component-based* model can significantly reduce the computational load. In the notation, a component-based model has three inputs: state variable (either dense or sparse), its index, and the index of the output state. More specifically,

$$\underline{x}(k) = \mathcal{M}(\underline{x}(k-1); \mathcal{I}_1; \mathcal{I}_2), \tag{5}$$

where $\mathcal{I}_1$ is the index set of the sparse vector $\underline{x}(k-1)$ and $\mathcal{I}_2$ is the index set of $\underline{x}(k)$. The model evaluates only the entries with indices in $\mathcal{I}_2$, setting all other entries as zeros. The indices in $\mathcal{I}_2$ represent those entries in $x(k)$ that are most sensitive to the variation of the entries in $x(k-1)$ with indices in $\mathcal{I}_1$. For instance, a discretization of PDE using finite difference results in a model, $\mathcal{M}$, such that each entry in $x(k)$ is sensitive only to the variation of its adjacent entries in $x(k-1)$. If the input vector, $x(k-1)$, is dense, we omit $\mathcal{I}_1$ in the notation, i.e.

**Table 1** Notations

| Notation | Definition | Notation | Definition |
|---|---|---|---|
| $x$ | State variable | $y$ | Observation variable |
| $\mathcal{M}$ | Model function | $\mathcal{H}$ | Observation operator |
| $n$ | State space dimension | $t = 1, 2, \ldots$ | (Discrete) time variable |
| $x_i^\sigma$ | $i$th $\sigma$-point | | |
| $x_i^b$ | $i$th background state | $y_i^b$ | Output of observation operator $\mathcal{H}(x_i^b)$ |
| $\bar{x}^b$ | Average of $x_i^b$ | $\bar{y}^b$ | Average of $y_i^b$ |
| $P^b$ | Background error covariance | | |
| $x^a$ | Analysis—state vector | $P^a$ | Analysis–error covariance |

$$\underline{x}(k) = \mathcal{M}(x(k-1); \mathcal{I}), \tag{6}$$

where $\mathcal{I}$ is the same as $\mathcal{I}_2$ in (5).

Additions and multiplications of vectors/matrices, in both dense and sparse formats, are carried out in their original spaces, $\mathbb{R}^n$ or $\mathbb{R}^{n\times n}$. For instance,

$$\underline{P}\,\underline{x} \text{ or } \underline{P}x$$

are both evaluated using matrix multiplication in $\mathbb{R}^n$ in the dense format. If one needs the value at only a sparse set of locations, it is denoted by another underbar mapping

$$\underline{(\underline{P}x)}$$

Similarly, the summation of a dense vector and a vector in sparse format makes sense. For instance,

$$\underline{x} + x$$

is a vector in $\mathbb{R}^n$ in which all entries of $x$ is unchanged except that those with indices in $\mathcal{I}$ are doubled. A new operation, called merging, between a sparse vector and a dense vector is defined as follows,

$$z = \underline{x} \triangleright w, \begin{cases} i\text{th component of } z = i\text{th component of } x, & \text{if } i \in \mathcal{I}. \\ i\text{th component of } z = i\text{th component of } w, & \text{if } i \notin \mathcal{I}. \end{cases} \tag{7}$$

A summary of notations is listed in the following Table 1.

## 3.2 UKF

The unscented Kalman filter has been increasingly popular in engineering applications since its introduction about twenty years ago (Julier et al. 2000; Julier and Uhlmann 2004). In a UKF, the error covariance is propagated with the dynamics using a set of vectors, or $\sigma$-points denoted by $x^\sigma$. Their definition is given in (8)–(9). The $\sigma$-points are computed at each time step using a square root of the error covariance. In most UKF applications, $\sigma$-points are computed using either Cholesky factorization or matrix diagonalization. In the notation, a variable with a superscript '$a$', such as $x^a$, represents the *analysis* value of the variable, i.e., the updated value based on observations. A variable with a superscript '$b$', such as $y^b$, represents the background, i.e., the propagated value of analysis using the system model. The algorithm is summarized as follows. At $t = k - 1$, suppose we have the analysis and error covariance as well as its square root

$$\begin{aligned} &x^a(k-1),\ \ P^a(k-1),\\ &X^a(k-1) = \sqrt{(n+\kappa)P^a(k-1)}, \end{aligned} \tag{8}$$

where $\kappa$ is a scaling factor for the fine tuning of the higher order moments of the approximation error (Julier et al. 2000). How to tune the value of $\kappa$ for a sparsity-based UKF is an open problem that needs further study. In this chapter, $\kappa = 0$ is used in all examples. A set of $\sigma$-points is generated as follows,

$$\begin{aligned} &x_0^\sigma(k-1) = x^a(k-1),\\ &x_i^\sigma(k-1) = x^a(k-1) + X_i^a(k-1),\ 1 \le i \le n,\\ &x_i^\sigma(k-1) = x^a(k-1) - X_i^a(k-1),\ n+1 \le i \le 2n. \end{aligned} \tag{9}$$

where $X_i^a(k-1)$ is the $i$th column vector of $X^a(k-1)$. The next step is to propagate the $\sigma$-points, which represent the background at $t = k$. For simplicity of notations, the time variable '$k$' in the $k$th time-step is omitted.

$$\begin{aligned} &x_i^b = \mathcal{M}(x_i^\sigma(k-1)), \quad y_i^b = \mathcal{H}(x_i^b), \qquad 0 \le i \le 2n,\\ &\bar{x}^b = \sum_{i=0}^{2n} w_i x_i^b, \qquad\quad \bar{y}^b = \sum_{i=0}^{2n} w_i y_i^b, \end{aligned} \tag{10}$$

where the weights are defined as follows

$$w_0 = \frac{\kappa}{n+\kappa},\ \ w_i = \frac{1}{2(n+\kappa)}, \tag{11}$$

for $i = 1, 2, \ldots, 2n$. Define the variations

$$X_i^b = x_i^b - \bar{x}^b,\ \ Y_i^b = y_i^b - \bar{y}^b. \tag{12}$$

The background covariances are

$$
\begin{aligned}
P^b &= \sum_{i=0}^{2n} w_i X_i^b (X_i^b)^T + Q,\\
P_{xy} &= \sum_{i=0}^{2n} w_i X_i^b (Y_i^b)^T,\\
P_{yy} &= \sum_{i=0}^{2n} w_i Y_i^b (Y_i^b)^T + R.
\end{aligned}
\tag{13}
$$

The Kalman gain, $K$, satisfies the following equation,

$$
K P_{yy} = P_{xy}. \tag{14}
$$

The analysis is updated as follows

$$
\begin{aligned}
x^a &= \bar{x}^b + K(y_o - \bar{y}^b),\\
P^a &= P^b - K(P_{xy})^T,
\end{aligned}
\tag{15}
$$

where $y_o$ is the observation at $t = k$. This completes one iteration of the filter. For the next step, $t = k + 1$, go back to (8) replacing the analysis by the updated value of $x^a$ and $P^a$.

### 3.3 Sparse-UKF

The square root factorization of a matrix is not unique. For large and sparse matrices, various algorithms and their implementations on different computing platforms have been studied for many years. The literature can be traced back to the early days of electronic computers (Davis et al. 2016). In the case of Cholesky factorization, the square root of a sparse matrix is still sparse, although the computation may require larger amounts of processor memory than the original matrix (Davis 2006; Rozin and Toledo 2005).

A dense error covariance is intractable in computation for global models used in NWP. In the following approach, we assume that $P$ and $\sqrt{P}$ are approximately sparse. In the algorithm, they are replaced by their sparse approximations, $\underline{P}$ and $\underline{(\sqrt{\underline{P}})}$. Their sparsity index sets are denoted by $\mathcal{I}$ and $\mathcal{I}^\sigma$, respectively. When propagating the $\sigma$-points using a component-based model, only a sparse subset of the elements is computed. The indices of the subset form an index set, $\mathcal{I}^b$. How to determine the index sets for sparse vectors and matrices is discussed later in this section

**Algorithm I** (sparse-UKF)
Given the initial analysis,

$$x^a(k-1),\ \underline{P}^a(k-1). \tag{16}$$

**Step 1**. $\sigma$-points and forecast

$$\underline{X}^a(k-1) = \underline{\sqrt{(n+\kappa)\underline{P}^a(k-1)}},\ \text{sparsity index set}\,\mathcal{I}^\sigma \tag{17}$$

For $i = 0$,

$$x_0^b = \mathcal{M}(x^a(k-1)), \quad y_0^b = \mathcal{H}(x_0^b). \tag{18}$$

For $i = 1, 2, 3, \ldots, 2n$,

$$\begin{aligned} x_i^\sigma(k-1) &= x^a(k-1) + \underline{X}_i^a(k-1), && 1 \le i \le n,\\ x_i^\sigma(k-1) &= x^a(k-1) - \underline{X}_i^a(k-1), && n+1 \le i \le 2n.\\ \underline{x}_i^b &= \mathcal{M}(x_i^\sigma(k-1); \mathcal{I}_i^b), \qquad y_i^b = \mathcal{H}(\underline{x}_i^b \triangleright x_0^b), && 1 \le i \le 2n. \end{aligned} \tag{19}$$

**Step 2**. Background covariances

$$\bar{x}^b = w_0 x_0^b + \sum_{i=1}^{2n} w_i(\underline{x}_i^b \triangleright x_0^b),\ \bar{y}^b = \sum_{i=0}^{2n} w_i y_i^b \tag{20}$$

$$\begin{aligned} \underline{P}^b &= w_0\underline{(x_0^b - \bar{x}^b)(x_0^b - \bar{x}^b)^T}\\ &\quad + \sum_{i=1}^{2n} w_i \underline{(\underline{x}_i^b \triangleright x_0^b - \bar{x}^b)(\underline{x}_i^b \triangleright x_0^b - \bar{x}^b)^T} + \underline{Q}, \quad \text{sparsity index set}\,\mathcal{I},\\ P_{xy} &= w_0(x_0^b - \bar{x}^b)(y_i^b - \bar{y}^b)^T + \sum_{i=1}^{2n} w_i(\underline{x}_i^b \triangleright x_0^b - \bar{x}^b)(y_i^b - \bar{y}^b)^T,\\ P_{yy} &= \sum_{i=0}^{2n} w_i(y_i^b - \bar{y}^b)(y_i^b - \bar{y}^b)^T + R. \end{aligned} \tag{21}$$

**Step 3**. Kalman gain and analysis

$$\begin{aligned} K P_{yy} &= P_{xy},\\ x^a &= \bar{x}^b + K(y_o - \bar{y}^b),\\ \underline{P}^a &= \underline{P}^b - \underline{K(P_{xy})^T} + \gamma I, \quad \text{sparsity index set}\,\mathcal{I}. \end{aligned} \tag{22}$$

The constant term $\gamma I$ in (22) is a diagonal matrix. The value of $\gamma$ is selected so that $\underline{P^a}$ is positive definite, which is guaranteed if $\gamma$ is larger than the smallest negative eigenvalue of

$$\underline{P}^b - \underline{K(P_{xy})^T}. \tag{23}$$

If this matrix is positive definite, then $\gamma = 0$. In the case that a fixed lower bound of eigenvalues is unknown, the value of $\gamma$ can be adaptively changed in every cycle depending on the smallest negative eigenvalue of (23). Numerical algorithms of finding the smallest eigenvalue for high dimensional matrices is needed for the determination of $\gamma$. Studies about this problem is out the scope of this paper. A survey on this topic can be found in Davidson (1989).

How to determine the index sets for sparse vectors/matrices? This is a problem for which we do not have a complete answer. The selection of $\mathcal{I}$ for $\underline{P}^a$ is a trade-off between the computational cost and the approximation accuracy. If $|\mathcal{I}|$ is small (highly sparse), $\underline{P}^a$ may not be a good approximation of $P^a$ because too many nonzero entries are set to zero; if $|\mathcal{I}|$ is large, it increases the computational cost. The sparsity index set, $\mathcal{I}^\sigma$, of $(\underline{\sqrt{\underline{P}^a}})$ is determined by the sparsity of $\sqrt{\underline{P}^a}$. If the square root is the Cholesky factorization, this matrix is already sparse. The number of nonzero entries in $\sqrt{\underline{P}^a}$ is larger than the size of $\mathcal{I}$ (Rozin and Toledo 2005). However, one may use a smaller set as $\mathcal{I}^\sigma$ to speed up the computation. Once again, this is a trade-off between computational cost and accuracy. In (19), $\mathcal{I}_i^b$ is the index set of $\underline{x}_i^b$, which is the propagation of the $i$th $\sigma$-point. The indices in $\mathcal{I}_i^b$ represent those entries in $x_i^b$ that are most sensitive to the variation of the entries of $x^\sigma(k-1)$ with indices in $\mathcal{I}_i^\sigma$. Or equivalently, $\underline{x}_i^b$ contains those entries of $x_i^b$ that have relatively large change when the value of $\underline{X}_i^a$ is changed. In general, these sparsity index sets are different from each other. However, as a means of reducing computational loads, we may use one index set for all three, $\mathcal{I}_i$, $\mathcal{I}_i^\sigma$ and $\mathcal{I}_i^b$. This idea is tested in the next section on a Lorenz-96 model.

In the sparse-UKF, the assumption is that $P^a$ can be approximated by a sparse matrix $\underline{P}^a$. Although the $\sigma$-points in the algorithm play a similar role as that of ensembles in EnKF, using sparse-UKF one can avoid the problem of rank deficiency. For systems with very high dimensions, the number of ensemble members used in an EnKF is much smaller than the dimension. As shown in Fig. 2 (left plot), the narrow and tall matrix of ensemble vectors makes EnKF fundamentally a rank deficient approach. In contrast, the block diagonal matrix $\underline{P}^a$ shown in Fig. 2 (middle plot) as a sparse approximation of $P^a$ has full rank.

The computational load required by (19) in Step 1 is extremely high if full state vectors are computed. Thanks to the sparsity, we only need to compute the entries with indices in $\mathcal{I}^b$. For a sparse-UKF to be successful for high dimensional problems, it is critical to have component-based numerical models so that only the entries with indices in $\mathcal{I}^b$ are computed; and most entries of the state vector are not evaluated at all. It is also important to point out that individual terms for $i = 1, 2, \ldots, 2n$ in (19), (20) and (21) can be computed independent of each other, making the computation perfectly parallel. Because matrices of covariance are symmetric, the memory size and I/O usage for the computation of covariance can be significantly reduced. For instance, the number of nonzero entries in the upper half of $\underline{P}^a$ is less than or equal to

$$n\left(\frac{N_{sp}-1}{2}+1\right) \tag{24}$$

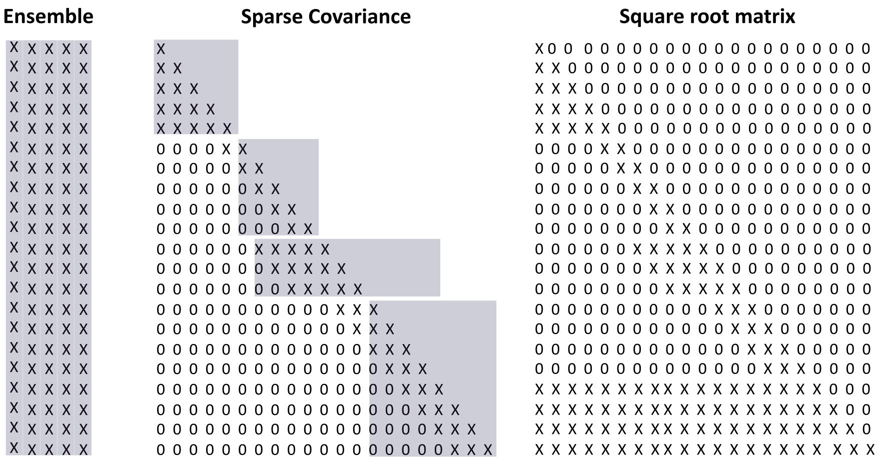


**Fig. 2** Patterns of ensemble vectors and sparse error covariances

If $N_{sp}$ is an integer close to the ensemble size of an EnKF, then (24) is smaller than the number of entries in the ensemble matrix, which is dense and nonsymmetric. Shown in Fig. 2 (middle and right plots) are the sparsity patterns of $\underline{P}^a$ and $\sqrt{\underline{P}^a}$ that we find in some examples. Note that the number of nonlinear entries in each column may vary. An advantage of the sparse-UKF is the capability of easily assigning different sparsity to different columns in $\underline{P}^a$ by using the index sets $\mathcal{I}_i$, $\mathcal{I}_i^\sigma$ and $\mathcal{I}_i^b$.

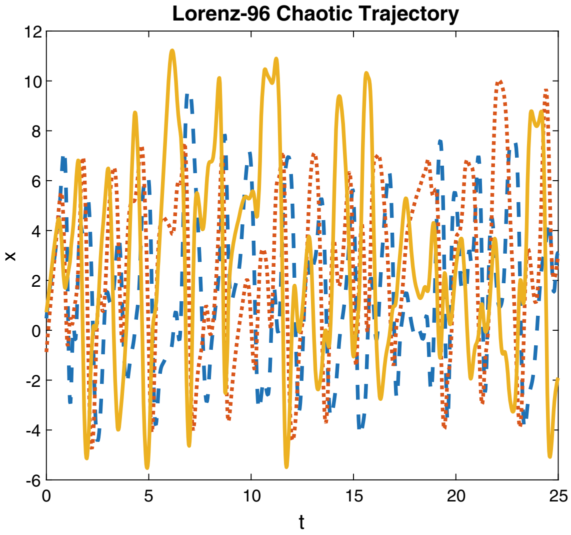
### *3.4 Lorenz-96 Model*

In this section, we use a Lorenz-96 model that was first introduced in Lorenz (1996) to test the performance of the sparse-UKF. Consider

$$\begin{aligned} \frac{dx_i}{dt} &= (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \quad i = 1, 2, \ldots, n,\\ x_{n+1} &= x_1,\\ n &= 40,\\ \Delta t &= 0.025,\\ F &= 8. \end{aligned} \tag{25}$$

The system has chaotic trajectories as shown in Fig. 3, a plot of $x_1(t), x_2(t), x_3(t)$. The simulations are conducted based on a 4th-order Runge-Kutta discretization. The trajectories are used as the ground truth. The sparsity pattern for $\underline{P}^a$ and $\sqrt{\underline{P}^a}$ are assumed to be centered along the diagonal line with a fix length. The total number of nonzero entries in each column is $N_{sp}$. We would like to point out that the sparse

**Fig. 3** A chaotic trajectory of the Lorenz-96 model, $x_1$(solid), $x_2$(dash), $x_3$(dot)



matrices are approximations of the true error covariance and its square root. The true sparsity pattern of $\sqrt{\underline{P}^a}$ is, in fact, different from that of $\underline{P}^a$. In the approximation, however, we ignore the difference and use the same sparsity pattern for both. This idea of simplifying index sets works fine for the Lorenz-96 model. A systematic way of choosing the sparsity pattern for $\sqrt{\underline{P}^a}$ based on given $\underline{P}^a$ is an open problem that needs further study.

The numerical experimentation is based on $N = 1000$ uniformly distributed random initial states in $[-1\ 1]$. The time step size is $\Delta t = 0.025$. The total number of time steps for each simulation is $N_t = 4000$. The number of observations at any given time is $m = 20$, i.e. every other state variable is measured,

$$y(k) = \left[ x_1(k)\ x_3(k)\ x_5(k)\ \cdots\ x_{39}(k) \right]^T. \tag{26}$$

The observational error has the Gaussian distribution. Its covariance is $R = I$, the identity matrix. The initial background error covariance is $P^b(0) = 0.2I$. The estimation error is defined by the following RMSE

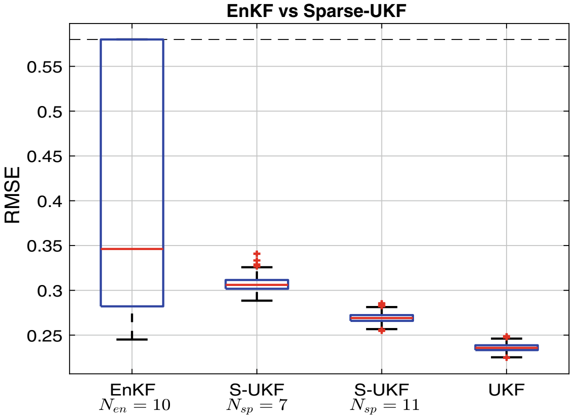
$$RMSE = \sqrt{\frac{1}{n(N_t+1)} \sum_{k=0}^{N_t} ||x^a(k) - x^{truth}(k)||_2^2}. \tag{27}$$

For comparison, an EnKF is also applied to the same data set. The localization radius is $\rho = 4$ and the inflation factor is $\sqrt{1.08}$. A full scale UKF based on dense error covariance is applied as the best estimator in the study. The number of nonzero entry evaluations in the computation of $\underline{P}^a$ and $\sqrt{\underline{P}^a}$, an indicator of computational load, is

**Table 2** Summary of simulation results

| Filter | Size | Nonzero | Entries | Error | Error | Error |
|---|---|---|---|---|---|---|
| | | Entries | EVAL | Median | Mean | STD |
| EnKF | $N_{ens} = 10$ | 400 in ensemble | 400 | 0.3462 | 1.0741 | 1.0652 |
| S-UKF | $N_{sp} = 7$ | 160 in $\underline{P}^a$ | 600 | 0.3061 | 0.3067 | 0.0071 |
| S-UKF | $N_{sp} = 11$ | 240 in $\underline{P}^a$ | 920 | 0.2691 | 0.2691 | 0.0048 |
| UKF | Full covariance | 820 in $P^a$ | 3200 | 0.2358 | 0.2360 | 0.0039 |

**Fig. 4** Boxplot of RMSE



$$2nN_{sp} + n \tag{28}$$

Reducing the number of entries being evaluated, such as using a smaller set of $\sigma$-points, and testing the impact of Cholesky factorization on the efficiency of UKF are ongoing research topics not addressed in this chapter.

Shown in Table 2, the EnKF has $N_{ens} = 10$ ensemble vectors with a total of 400 nonzero entries. In comparison, the sparse-UKFs with $N_{sp} = 7$ and 11 have much smaller numbers of nonzero entries to be evaluated and stored in memory. A smaller number is desirable because it implies reduce I/O load and the amount of memory usage. In terms of computational load, the number of entry evaluations for the sparse-UKFs are higher. This is mainly due to the propagation of the $2n$ $\sigma$-points. Studies show that reducing the number of $\sigma$-points to $n$ is possible. However, its impact on the estimation accuracy has to be studied case by case, which is beyond the scope of this chapter. In the columns under error median and mean, the numbers show that both sparse-UKFs achieve more accurate estimation than EnKF. The most significant advantage of sparse-UKFs are the small variation of estimation error. In Table 2, the error standard deviation of the sparse-UKFs are 0.0071 and 0.0048, which is in sharpe contrast to 1.0652 of EnKF. The error of EnKF has large variation due to the method's high dependency on the subspace in which the ensembles are selected. This problem does not exist for the sparse-UKF because $\underline{P}^a$ has full rank. The variation of errors is clearly shown in the boxplot in Fig. 4. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points the algorithm considers to be not outliers, and the outliers are plotted individually. For comparison, the error boxplot of the full size UKF is included in Fig. 4.

# 4 Progressive-EKF

In a sparse-UKF, the $\sigma$-points are computed by taking a square root of the error covariance, such as the Cholesky factorization. In this section, we propose a progressive algorithm of approximating error covariance without taking square roots.

## 4.1 *Basic Ideas*

The main assumption for this algorithm is the following progressive relationship

$$M_{k-1}P^a(k-1)M_{k-1}^T = P^a(k-1) + \Delta P^b, \tag{29}$$

where $\Delta P^b$ is assumed to be small. In (29), $M_{k-1}$ is the Jacobian of $\mathcal{M}$ at $x^a(k-1)$. Similarly, the Jacobian of $\mathcal{H}$ is $H_k$. To estimate $\Delta P^b$, assume

$$M_{k-1} = I + \Delta M_{k-1}. \tag{30}$$

where we assume that $\Delta M_{k-1}$ is small. If the system model is based on the discretization of a differential equation with a small time step size, then

$$\mathcal{M}(x(k-1)) = x(k-1) + O(\Delta t^\alpha), \quad \alpha > 1. \tag{31}$$

The Jacobian of $O_{k-1}(\Delta t^\alpha)$ in space variables is expected to have small value if $\Delta t$ is small, which makes (30) a reasonable assumption. Then we have

$$\begin{aligned}
&M_{k-1}P^a(k-1)M_{k-1}^T \\
&= (I + \Delta M_{k-1})P^a(k-1)(I + \Delta M_{k-1}^T) \\
&= P^a(k-1) + \Delta M_{k-1}P^a(k-1) + \left(\Delta M_{k-1}P^a(k-1)\right)^T \\
&\quad + \Delta M_{k-1}P^a(k-1)\Delta M_{k-1}^T \\
&\approx P^a(k-1) + \Delta M_{k-1}P^a(k-1) + \left(\Delta M_{k-1}P^a(k-1)\right)^T .
\end{aligned} \tag{32}$$

This is in consistent with (29). It can be computed using a tangent linear model. Or it can be approximated using the dynamical model

$$\begin{aligned}
&M_{k-1}P^a(k-1)M_{k-1}^T \\
&= (I+\Delta M_{k-1})P^a(k-1)(I+\Delta M_{k-1}^T) \\
&\approx \big(\mathcal{M}(x(k-1)+\delta P^a(k-1)) - \mathcal{M}(x(k-1))\big)/\delta \\
&+ \big(\mathcal{M}(x(k-1)+\delta P^a(k-1)) - \mathcal{M}(x(k-1))\big)^T/\delta - P^a.
\end{aligned} \tag{33}$$

where $\delta > 0$ is the step size of a finite difference approximation of $\Delta M_{k-1}P^a$. Its value should be determined depending on the numerical model and its linearization. In (33), a vector and matrix summation is a new matrix resulting from adding the vector to every column in the matrix. Applying an operator to a matrix is to apply the operator to every column in the matrix.

## 4.2 Progressive-EKF

The column vectors in the matrices in (32) and (33) are sparse. However, the number of column vectors equals $n$, which can be as high as $10^6$–$10^7$ for some atmospheric models. Applying a full model to all the vectors is impractical because of the high computational and I/O loads. Similar to the idea that we used in sparse-UKF, we approximate the error covariance using a given sparsity, i.e., only a small portion of the entries in each column vector is evaluated. Evaluating the entire state vector is unnecessary. This is the reason we need a component-based model. Then the algorithm of progressive-EKF is summarized as follows.

**Algorithm II** (progressive-EKF)
Given the initial analysis at $t = k-1$,

$$x^a(k-1) \text{ and } \underline{P}^a(k-1). \tag{34}$$

**Step 1**. Forecast

$$\begin{aligned}
x^b &= \mathcal{M}(x^a(k-1)), \\
y^b &= \mathcal{H}(x^b).
\end{aligned} \tag{35}$$

**Step 2**. Background error covariance

$$\begin{aligned}
\underline{P}^b = &\underline{\big(\mathcal{M}\left(x^a(k-1)+\delta\underline{P}^a(k-1),\mathcal{I}\right) - x^b\big)}/\delta \\
&+ \underline{\big(\mathcal{M}\left(x^a(k-1)+\delta\underline{P}^a(k-1),\mathcal{I}\right) - x^b\big)^T}/\delta - \underline{P}^a + Q.
\end{aligned} \tag{36}$$

**Step 3**. Kalman gain and analysis

$$\begin{aligned} K &= \underline{P}^b H_k^T (H_k \underline{P}^b H_k^T + R)^{-1}, \\ x^a &= x^b + K(y_o - y^b), \\ \underline{P}^a &= (I - KH_k)\underline{P}^b. \end{aligned} \tag{37}$$

Different from the sparse-UKF, this algorithm avoids the computation of matrix square roots. However, the algorithm requires that $\Delta P^b$ in (29) can be approximated effectively. From (31), the method is expected to work better for a small time step-size. If $\Delta t$ is large, $\Delta M_{k-1}$ in (30) may not be small enough. A remedy is to use a refined step-size in an inner-loop computation. More specifically, the discrete model is a discretization of a continuous-time model. The discrete time moment $k-1$ corresponds to the continuous time moment $(k-1)\Delta t$. We refine the step size by dividing the time interval into $n_p$ subintervals. In our examples, we choose $n_p = 2$. The refined time steps are

$$(k-1)\Delta t, (k-1)\Delta t + \frac{\Delta t}{n_p}, \ldots, (k-1)\Delta t + s\frac{\Delta t}{n_p}, \ldots, k\Delta t, \qquad 0 \le s \le n_p \tag{38}$$

For the inner loop, one can compute a sequence of backgrounds, $\tilde{x}^b(s)$.

$$\begin{aligned} t_s &= (k-1)\Delta t + s\frac{\Delta t}{n_p}, \\ \tilde{x}^b(s) &= \tilde{\mathcal{M}}_{t_s}(x^a(k-1)),\ s = 1, 2, \ldots, n_p. \end{aligned} \tag{39}$$

where $\tilde{\mathcal{M}}_{t_s}$ represents the refined model function in the time interval from $t = (k-1)\Delta t$ to $t = t_s$. In Step 2, repeat (36) $n_p$ times along the sequence of background states, $\tilde{x}^b(s)$, without adding $Q$ until the last round. This refined Step 2 increases the computational load, while improving the accuracy of the progressive estimation.

## 4.3 Examples

In the following, we apply the progressive-EKF to the Lorenz-96 model using the same parameters given in (25). The analysis is based on the simulation data using $N = 1000$ random initial states, in which the value of each state variable is uniformly distributed in $[-1, 1]$. The error covariance is approximated using sparsity matrices with $N_{sp} = 7, 11, 17$. For $N_{sp} = 11$, we tested the idea of refining step-size using $n_p = 1$ and $n_p = 2$. The results are summarized in Table 3. The boxplots of error variation are shown in Fig. 5. Comparing to EnKF, the error variations of the progressive-EKFs are significantly smaller. If $N_{sp} = 7$, which is smaller than the ensemble size $N_{ens} = 10$, the median value of estimation error is larger than that of the EnKF. The median error for $N_{sp} = 11$ is comparable to that of the EnKF. If a refined step-size in (39) is applied, for instance $n_p = 2$, the median estimation error is further reduced. Comparing to the performance of the sparse-UKF in Table 2, the

**Table 3** Summary of simulation results

| Filter | Size | Nonzero | Entries | Error | Error | Error |
|---|---|---|---|---|---|---|
| | | Entries | EVAL | Median | Mean | STD |
| EnKF | $N_{ens} = 10$ | 400 in ensemble | 400 | 0.3462 | 1.0741 | 1.0652 |
| P-EKF | $N_{sp} = 7$<br>$N_p = 1$ | 160 in $\underline{P}^a$ | 320 | 0.3845 | 0.3846 | 0.0055 |
| P-EKF | $N_{sp} = 11$<br>$N_p = 1$ | 240 in $\underline{P}^a$ | 480 | 0.3455 | 0.3458 | 0.0050 |
| P-EKF | $N_{sp} = 11$<br>$N_p = 2$ | 240 in $\underline{P}^a$ | $480 \times 2$ | 0.3041 | 0.3041 | 0.0044 |
| P-EKF | $N_{sp} = 17$<br>$N_p = 3$ | 360 in $\underline{P}^a$ | $720 \times 2$ | 0.2872 | 0.2873 | 0.0046 |

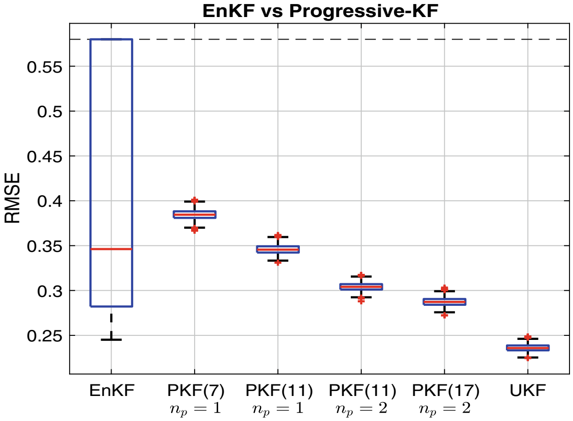


**Fig. 5** Boxplot of RMSE. For Prograssive-KF, $N_{sp}$ = 7, 11, and 17

error variations are similar. However, the estimation error of the sparse-UKF has a smaller median in all cases. For example, to achieve a similar performance as the sparse-UKF when $N_{sp} = 11$, one has to use a larger sparsity index $N_{sp} = 17$ for the progressive-EKF.

## 5 Conclusions

Two Kalman type filters, sparse-UKF and progressive-EKF, based on sparse error covariances are introduced. They are tested using the Lorenz-96 model with 40 state variables and chaotic trajectories. Both algorithms share the same basic idea: the error covariance is approximated using a sparse matrix. Thanks to the sparsity, the required memory size is significantly reduced. The symmetry of the error covariance can potentially reduce the I/O load. The analysis error covariance can be updated as a sparse matrix in each cycle using a deterministic process, either a square root matrix or a progressive algorithm. The updated sparse matrix is then used as the background error covariance for the next cycle. Relative to EnKFs, the main advantage of the proposed methods is that the estimation process do not need an ensemble; and the error covariance has a full rank. The algorithms do not suffer issues of rank deficiency as in EnKFs. As a result, the variation of analysis error is constantly small in all examples. Techniques of localization and covariance inflation are unnecessary. Relative to 4D-Var methods, the proposed algorithms are highly parallel in computation. They provide not only the estimate of states but also the analysis error covariance. For the purpose of scalability, we suggest that the proposed methods are applied with component-based numerical models. From the examples, the sparse-UKF has better accuracy than the progressive-EKF. On the other hand, the progressive-EKF is a simple algorithm that avoids taking square roots of large matrices, provided that the progressive approximation of error covariance is adequately accurate. The limited number of examples in this chapter is not enough for drawing a comprehensive comparison between the two filters. More numerical experimentations and further study of the methods using different types of system models are main topics of our future work.

## References

Davidson ER (1989) Super-matrix methods. Comput Phys Commun 53(1–3):49–60

Davis TA (2006) Direct methods for sparse linear systems. SIAM

Davis T, Rajamanickam S, Sid-Lakhdar WM (2016) A survey of direct methods for sparse linear systems. Acta Numer 25:383–566

Houtekamer PL, Zhang F (2016) Review of the ensemble Kalman filter for atmospheric data assimilation. Mon Weather Rev 144:4489–4532

Julier S, Uhlmann J, Durrant-Whyte HF (2000) A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Trans Autom Control 45(3):477–482

Julier SJ, Uhlmann JK (2004) Unscented filtering and nonlinear estimation. Proc IEEE 92(3):401–422

Kang W, Xu L (2021) Some quantitative characteristics of error covariance for Kalman filters. Tellus A: Dyn Meteorol Ocenogr 73(1):1–19

Lorenz E (1996) Predictability—a problem partly solved. In: Seminar on predictability, vol I. ECMWF

Rozin E, Toledo S (2005) Locality of reference in sparse Cholesky factorization methods. Electron Trans Numer Anal 21:81–106

Xu L, Rosmond R, Daley R (2005) Development of NAVDAS-AR: formulation and initial tests of the linear problem. Tellus 57A:546–559