

Ensemble propagation and continuous matrix factorization algorithms

Kay Bergemann* Georg Gottwald† Sebastian Reich‡

May 4, 2009

Abstract

We consider the problem of propagating an ensemble of solutions and its characterization in terms of its mean and covariance matrix. We propose differential equations that lead to a continuous matrix factorization of the ensemble into a generalized singular value decomposition (SVD). The continuous factorization is applied to ensemble propagation under periodic rescaling (ensemble breeding) and under periodic Kalman analysis steps (ensemble Kalman filter). We also use the continuous matrix factorization to perform a re-orthogonalization of the ensemble after each time-step and apply the resulting modified ensemble propagation algorithm to the ensemble Kalman filter. Results from the Lorenz-96 model indicate that the re-orthogonalization of the ensembles leads to improved filter performance.

Keywords. Data assimilation, ensemble Kalman filter, ensemble breeding, singular value decomposition, continuous matrix factorization.

1 Introduction

In many applications several independent realizations (an ensemble) $\mathbf{x}_i(t) \in \mathbb{R}^k$, $i = 1, \dots, m$, of a dynamical system are available and can be analyzed in terms of their mean

$$\bar{\mathbf{x}}(t) = \frac{1}{m} \sum_i \mathbf{x}_i(t) \quad (1)$$

and their deviations

$$\mathbf{x}'_i(t) = \mathbf{x}_i(t) - \bar{\mathbf{x}}(t), \quad i = 1, \dots, m. \quad (2)$$

One particularly useful information about the deviations is provided by a singular value decomposition (SVD) of the ensemble deviation matrix

$$\mathbf{X}'(t) = [\mathbf{x}'_1(t), \dots, \mathbf{x}'_m(t)] \in \mathbb{R}^{k \times m}. \quad (3)$$

In principle, a standard SVD of $\mathbf{X}'(t)$ can be performed at any desired moment in time t_n . However, if $t_n = n \Delta t$ and $\Delta t \rightarrow 0$, it seems advantageous to find a technique to “continuously” update a SVD from t_n to t_{n+1} instead of computing SVDs from scratch. Continuous SVD

*Universität Potsdam, Institut für Mathematik, Am Neuen Palais 10, D-14469 Potsdam, Germany

†School of Mathematics and Statistics, University of Sydney, NSW 2006, Australia

‡Universität Potsdam, Institut für Mathematik, Am Neuen Palais 10, D-14469 Potsdam, Germany

updates have been studied in the context of linear time-varying ordinary differential equations (ODEs) of the type

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} \quad (4)$$

and for $k = m$. A main application area is the computation of Lyapunov exponents of nonlinear dynamical systems. See, for example, Greene and Kim (1987); Wright (1992); Geist et al. (1990).

Here we instead consider ensembles generated by a nonlinear ODE

$$\dot{\mathbf{x}} = f(\mathbf{x}) \quad (5)$$

and are mainly interested in applications for which $m \ll k$. Such ensemble formulations are often used in meteorology (with typical values of $k \approx 10^9$ and $m \approx 10^2$ in operational weather forecasts) and other application areas in the context of ensemble prediction and ensemble Kalman filters (Kalnay, 2002; Evensen, 2006). Our approach consists of a continuous factorization of the ensemble deviation matrix \mathbf{X}' into a matrix product $\mathbf{Q}\mathbf{M}$, where $\mathbf{Q} \in \mathbb{R}^{k \times m}$, $\mathbf{M} \in \mathbb{R}^{m \times m}$. For general matrix-valued solutions of a linear ODE (4) one typically requests that such a factorization satisfies $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_m$ (Dieci and van Vleck, 1995; Bridges and Reich, 2001). However, in our situation, the deviations \mathbf{x}'_i are linearly dependent because of $\sum_i \mathbf{x}'_i = \mathbf{0}$. This requires a modified orthogonality constraint. Furthermore, given a SVD of $\mathbf{M}(t)$ at $t = t_n$, we also propose a simple iterative update to obtain the SVD of $\mathbf{M}(t)$ at $t = t_{n+1}$. We apply the proposed continuous SVD ensemble representation to the Lorenz-96 model (Lorenz, 1996; Lorenz and Emanuel, 1998).

The paper is organized as follows. In Section 2, we summarize the basic concepts of ensemble propagation. Continuous matrix factorizations in terms of differential equations for the matrix factors are discussed in Section 3. We first summarize basic results for linear differential equations and propose a new formulation for nonlinear ensemble propagation. Numerical implementations are discussed in Section 4. A numerically robust and efficient implementation is summarized in the Appendix. In Section 5, we apply the continuous factorization algorithm to the Lorenz-96 model both in the context of ensemble breeding (Toth and Kalnay, 1993, 1997) and ensemble Kalman filtering (Evensen, 2006). We also propose a modified ensemble update for ensemble transform Kalman filters (Bishop et al., 2001; Tippett et al., 2003; Wang et al., 2004), which is based on the continuous Kalman filter formulation (see, for example, Simon (2006)). We finally discuss an application of the matrix factorization in terms of modified ensemble propagation in Section 6. The basic idea is to "re-orthogonalize" the ensemble after each time-step. We demonstrate that such a modification of the ensemble propagation improves the performance of the ensemble transform Kalman filter applied to the Lorenz-96 model.

2 Ensemble propagation

We summarize the basic idea of an ensemble forecast (Evensen, 2006). Given a differential equation (5) in the state variable $\mathbf{x} \in \mathbb{R}^k$, we consider the simultaneous propagation of m independent solutions

$$\dot{\mathbf{x}}_i = f(\mathbf{x}_i), \quad i = 1, \dots, m, \quad (6)$$

which we write in more compact form as

$$\dot{\mathbf{X}} = f(\mathbf{X}) \quad (7)$$

with

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{k \times m}, \quad f(\mathbf{X}) = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_m)] \in \mathbb{R}^{k \times m}. \quad (8)$$

The matrix $\mathbf{X}(t)$ is called the ensemble at time t and $\mathbf{x}_i(t)$, $i = 1, \dots, m$, are its members.

The ensemble is typically analyzed in terms of its mean

$$\bar{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i = \mathbf{X}\mathbf{w}, \quad \mathbf{w} = \frac{1}{m} [1, \dots, 1]^T \in \mathbb{R}^m \quad (9)$$

and its ensemble deviation matrix

$$\mathbf{X}' = \mathbf{X} - \bar{\mathbf{x}}\mathbf{e}^T = \mathbf{X}\mathbf{T}, \quad (10)$$

where $\mathbf{T} \in \mathbb{R}^{m \times m}$ is a constant projector matrix defined by

$$\mathbf{T} = \mathbf{I}_m - \mathbf{w}\mathbf{e}^T, \quad \mathbf{e} = [1, \dots, 1]^T \in \mathbb{R}^m. \quad (11)$$

The ensemble deviation matrix $\mathbf{X}'(t)$ gives rise to the ensemble covariance matrix

$$\mathbf{P}(t) = \frac{1}{m-1} \mathbf{X}'(t) [\mathbf{X}'(t)]^T. \quad (12)$$

Note that the ensemble mean $\bar{\mathbf{x}}$ is generally not a solution of (5) since

$$f(\mathbf{X})\mathbf{w} \neq f(\bar{\mathbf{x}}), \quad (13)$$

in general.

Provided that $\mathbf{X}' = \mathbf{X}\mathbf{T}$ is sufficiently small, we may apply Taylor expansion and approximate (7) by

$$\dot{\mathbf{X}} = f(\bar{\mathbf{x}})\mathbf{e}^T + \mathbf{A}(t)\mathbf{X}\mathbf{T}, \quad (14)$$

where $\mathbf{A}(t) = Df(\bar{\mathbf{x}}(t))$ denotes the Jacobian matrix of f at $\bar{\mathbf{x}}(t)$. Note that the evolution equation for the ensemble deviation matrix \mathbf{X}' is now given by

$$\dot{\mathbf{X}}' = \mathbf{A}(t)\mathbf{X}'. \quad (15)$$

Numerically, we are often given a one-step method

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \phi(\mathbf{x}_n; \Delta t) \quad (16)$$

for the ODE (5), which can be generalized to the ensemble ODE formulation (7). We denote the resulting ensemble approximation at time-levels $t_n = n \Delta t$, $n \geq 0$, by

$$\mathbf{X}_n = \bar{\mathbf{x}}_n \mathbf{e}^T + \mathbf{X}'_n. \quad (17)$$

3 Continuous matrix factorization formulations

In many instances it is useful to have a SVD of the ensemble deviation matrix $\mathbf{X}' \in \mathbb{R}^{k \times m}$ available. Since $\mathbf{X}' = \mathbf{X}'\mathbf{T}$, we propose the following modified SVD representation:

$$\mathbf{X}' = \mathbf{Q}\mathbf{M}, \quad \mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (18)$$

where $\mathbf{Q} \in \mathbb{R}^{k \times m}$ satisfies $\mathbf{Q}^T \mathbf{Q} = \mathbf{T}$, $\mathbf{M} \in \mathbb{R}^{m \times m}$ is a regular (invertible) matrix satisfying

$$\mathbf{M}\mathbf{w} = \mathbf{w}, \quad (19)$$

and $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ denotes the standard SVD Golub and Loan (1996) of \mathbf{M} . Note that

$$\mathbf{X}'\mathbf{w} = \mathbf{Q}\mathbf{M}\mathbf{w} = \mathbf{Q}\mathbf{w} = \mathbf{0} \quad (20)$$

as desired. Furthermore, the columns of \mathbf{Q} span the same linear subspace as the deviations \mathbf{x}'_i , $i = 1, \dots, m$, while the diagonal entries (singular values) of the diagonal matrix $\mathbf{\Sigma}$ characterize the magnitude of the deviations. Note that, due to (19), one of the singular values is equal to one.

Let us assume that the ensemble deviation matrix \mathbf{X}' and its SVD (18) are known at time-level t_n . Then we wish to find a simple update which yields the SVD of \mathbf{X}' at the subsequent time-level t_{n+1} . It is the aim of this paper to suggest such an algorithm.

Note that the standard SVD of \mathbf{X}' would lead to a representation of type (18) with $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_m$ and $\mathbf{\Sigma}$ would contain zero as a singular value. In terms of our continuous formulations it is, however, advantageous to keep $\mathbf{\Sigma}$ invertible and \mathbf{Q} rank deficient. We will come back to this aspect in Section 3.2.

Our work is an extension of the Stiefel integrators proposed by Bridges and Reich (2001) and we summarize the basic ideas for linear systems first.

3.1 General linear problems

Popular matrix factorization methods include the QR, SVD, and polar factorization (Golub and Loan, 1996). In this paper, we rely on the polar factorization of non-square $k \times m$ matrices, $k > m$, and the SVD factorization of positive definite $m \times m$ matrices.

While the SVD factorization is standard, the polar factorization is less well-known and we first recall a number of basic results. A polar factorization of a square matrix \mathbf{X} is defined as the product \mathbf{QM} of an orthogonal matrix \mathbf{Q} and a symmetric non-negative matrix \mathbf{M} . If \mathbf{M} is regular ($\det \mathbf{M} \neq 0$), then the factorization is unique. Note that polar factorizations are linked to optimal transportation theory (Brenier, 1991). In linear algebra terms, the orthogonal polar factor \mathbf{Q} is a nearest unitary matrix to \mathbf{X} in the Frobenius norm. For general (non-singular) $m \times m$ matrices \mathbf{X} we may use the following polar factorization algorithm. Compute the SVD of $\mathbf{X}^T \mathbf{X}$, set \mathbf{M} equal to the square root of $\mathbf{X}^T \mathbf{X}$, and $\mathbf{Q} = \mathbf{X} \mathbf{M}^{-1}$. More generally, given a non-square matrix $\mathbf{X} \in \mathbb{R}^{k \times m}$, $k > m$, the polar factorization can be computed in several ways (Higham and Schreiber, 1990). The most straightforward is to first compute a QR decomposition of \mathbf{X} and to then find the polar factorization of the resulting upper triangular $m \times m$ matrix.

We now recall the continuous matrix factorization algorithms of Bridges and Reich (2001). Consider a linear time-dependent differential equation (15). Following the presentation of Bridges and Reich (2001), we do not distinguish between \mathbf{X} and \mathbf{X}' in this section, i.e., we formally set $\mathbf{T} = \mathbf{I}_m$ in (11) and, hence, do not assume that the mean of \mathbf{X}' vanishes. We wish to continuously factorize a general solution $\mathbf{X}(t) = \mathbf{X}'(t)$ into

$$\mathbf{X}(t) = \mathbf{Q}(t)\mathbf{M}(t), \quad (21)$$

where $\mathbf{Q} \in \mathbb{R}^{k \times m}$ and $\mathbf{M} \in \mathbb{R}^{m \times m}$. Hence we obtain

$$\dot{\mathbf{X}} = \dot{\mathbf{Q}}\mathbf{M} + \mathbf{Q}\dot{\mathbf{M}} = \mathbf{A}(t)\mathbf{Q}\mathbf{M} \quad (22)$$

and

$$\dot{\mathbf{Q}} = \mathbf{A}(t)\mathbf{Q} - \mathbf{Q}\mathbf{S}(t), \quad (23)$$

$$\dot{\mathbf{M}} = \mathbf{S}(t)\mathbf{M}, \quad (24)$$

where $\mathbf{S}(t) \in \mathbb{R}^{m \times m}$ is still to be determined by appropriate additional assumptions. For example, a continuous QR algorithm is obtained by choosing $\mathbf{S}(t)$ such that (i) $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_m$ and (ii) \mathbf{M} is upper triangular (Dieci and van Vleck, 1995). A continuous polar factorization was

instead suggested by Bridges and Reich (2001), i.e., (i) $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_m$ and (ii) \mathbf{M} is symmetric non-negative.

All these factorizations require that $\mathbf{Q} \in \mathbb{R}^{k \times m}$ is an element of the Stiefel manifold $V_m(\mathbb{R}^k)$ defined by

$$V_m(\mathbb{R}^k) = \{\mathbf{Q} \in \mathbb{R}^{k \times m} : \mathbf{Q}^T \mathbf{Q} = \mathbf{I}_m\}. \quad (25)$$

Stiefel manifolds are a generalization of the $(k-1)$ -dimensional unit sphere, which is obtained for $m=1$. If $k=m$, the Stiefel manifold becomes $O(k)$, the orthogonal group. A Stiefel manifold is a regular submanifold of $\mathbb{R}^{k \times m}$ of dimension $km - \frac{1}{2}m(m-1)$.

Let us now discuss the choice of $\mathbf{S}(t)$ in more detail. To do so we introduce the decomposition of \mathbf{S} into its symmetric part \mathbf{S}^{sym} and its skew-symmetric part \mathbf{S}^{skew} . We note $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_m$ implies that

$$\frac{d}{dt}(\mathbf{Q}^T \mathbf{Q}) = \mathbf{Q}^T \mathbf{A}(t) \mathbf{Q} + \mathbf{Q}^T \mathbf{A}(t)^T \mathbf{Q} - \mathbf{S}(t) - \mathbf{S}(t)^T = \mathbf{0} \quad (26)$$

provided

$$\mathbf{S}^{\text{sym}}(t) = \frac{1}{2} \mathbf{Q}^T [\mathbf{A}(t) + \mathbf{A}(t)^T] \mathbf{Q}. \quad (27)$$

Since (26) holds only for $\mathbf{Q} \in V_m(\mathbb{R}^k)$, we say that the Stiefel manifold $V_m(\mathbb{R}^k)$ is a weak invariant of (23)-(24) (Bridges and Reich, 2001).

There is a remaining freedom in the choice of the skew-symmetric part of $\mathbf{S}(t)$. Different choices of \mathbf{S}^{skew} lead to different paths on the Stiefel manifold and to different factorizations. For, example, one might define $\mathbf{S}^{\text{skew}}(t)$ such that $\mathbf{S}(t)$ becomes upper triangular, which then leads to a continuous QR algorithm (Dieci and van Vleck, 1995). Bridges and Reich (2001) instead set $\mathbf{S}^{\text{skew}}(t) = \mathbf{0}$. The resulting equations (23)-(24) lead to a polar factorization of solutions $\mathbf{X}(t) = \mathbf{Q}(t)\mathbf{M}(t)$ with initial conditions $\mathbf{M}(0) \in \mathbb{R}^{m \times m}$ and $\mathbf{Q}(0) \in \mathbb{R}^{k \times m}$ subject to $\mathbf{Q}(0)^T \mathbf{Q}(0) = \mathbf{I}_m$ and $\mathbf{M}(0)$ symmetric, positive-definite.

For general initial conditions with $\mathbf{M}(0)$ symmetric and non-negative, we have to use the remaining freedom in the skew-symmetric part of \mathbf{S} to keep the time evolution of \mathbf{M} symmetric, i.e.,

$$\dot{\mathbf{M}} = \dot{\mathbf{M}}^T. \quad (28)$$

Using the *ansatz* $\mathbf{S} = \mathbf{S}^{\text{sym}} + \mathbf{S}^{\text{skew}}$ with \mathbf{S}^{sym} given by (27), we obtain from (28) a linear equation for \mathbf{S}^{skew} of the form

$$\mathbf{S}^{\text{sym}} \mathbf{M} - \mathbf{M} \mathbf{S}^{\text{sym}} + \mathbf{S}^{\text{skew}} \mathbf{M} + \mathbf{M} \mathbf{S}^{\text{skew}} = \mathbf{0}, \quad (29)$$

which can be solved for \mathbf{S}^{skew} provided \mathbf{M} is non-singular.

Under appropriate conditions (Dieci and Elia, 2006), symmetric solutions $\mathbf{M}(t) \in \mathbb{R}^{m \times m}$ of (24) can be continuously factorized into a SVD, i.e.,

$$\mathbf{M}(t) = \mathbf{V}(t) \boldsymbol{\Sigma}(t) \mathbf{V}(t)^T, \quad (30)$$

where $\mathbf{V}(t)$ approaches a constant matrix as $t \rightarrow \infty$. The diagonal entries $\sigma_i(t)$, $i=1, \dots, m$ of $\boldsymbol{\Sigma}(t)$ are linked to the Lyapunov exponents of the system (15). Differential equations for the evolution of the diagonal entries σ_i , $i=1, \dots, m$, of $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$ and the evolution of the unitary matrix $\mathbf{V} \in O(m)$ can, for example, be found in Greene and Kim (1987); Wright (1992). An alternative ODE formulation with build-in error correction has been proposed by Baumann and Helmke (2003).

3.2 Nonlinear ensemble propagation

We now consider the nonlinear problem (7) and the continuous polar factorization of the ensemble deviation matrix $\mathbf{X}' = \mathbf{X}\mathbf{T}$ with the projector \mathbf{T} given by (11). We set $\mathbf{X}' = \mathbf{Q}\mathbf{M}$ with

$\mathbf{Q} \in \mathbb{R}^{k \times m}$ and $\mathbf{M} \in \mathbb{R}^{m \times m}$ as before. Since $\mathbf{X}'\mathbf{T} = \mathbf{X}'$, we replace $\mathbf{Q} \in V_m(\mathbb{R}^k)$ by

$$\mathbf{Q} \in \widehat{V}_m(\mathbb{R}^k) = \{\mathbf{Q} \in \mathbb{R}^{k \times m} : \mathbf{Q}^T \mathbf{Q} = \mathbf{T}, \mathbf{Q} = \mathbf{Q}\mathbf{T}\} \quad (31)$$

and request that \mathbf{M} satisfies

$$\mathbf{M}\mathbf{w} = \mathbf{w} \quad (32)$$

which makes one an eigenvalue of \mathbf{M} , and \mathbf{M} is invertible, in general. For \mathbf{M} symmetric, (32) implies that \mathbf{M} commutes with \mathbf{T} , i.e.,

$$\mathbf{M}\mathbf{T} = \mathbf{T}\mathbf{M}. \quad (33)$$

Since we are now dealing with a nonlinear ensemble propagation, eq. (23) needs to be replaced by

$$\dot{\mathbf{Q}} = f(\bar{\mathbf{x}}\mathbf{e}^T + \mathbf{Q}\mathbf{M})\mathbf{T}\mathbf{M}^{-1} - \mathbf{Q}\mathbf{S}(t), \quad (34)$$

$$\dot{\bar{\mathbf{x}}} = f(\bar{\mathbf{x}}\mathbf{e}^T + \mathbf{Q}\mathbf{M})\mathbf{w}. \quad (35)$$

The evolution equation for $\mathbf{M}(t)$ is still given by (24), the symmetric part of $\mathbf{S}(t)$ is now defined by

$$\mathbf{S}^{\text{sym}}(t) = \frac{1}{2} \left[\mathbf{Q}^T f(\mathbf{X})\mathbf{T}\mathbf{M}^{-1} + \mathbf{M}^{-T}\mathbf{T} [f(\mathbf{X})]^T \mathbf{Q} \right], \quad \mathbf{X} = \bar{\mathbf{x}}\mathbf{e}^T + \mathbf{Q}\mathbf{M}, \quad (36)$$

and, for symmetric $\mathbf{M}(t)$, the skew-symmetric part of $\mathbf{S}(t)$ is determined by (29).

Due to our specific choice of the generalized SVD decomposition (18), i.e., $\mathbf{M}\mathbf{w} = \mathbf{w}$ and $\mathbf{Q}\mathbf{w} = \mathbf{0}$, we have

$$\bar{\mathbf{x}}\mathbf{e}^T + \mathbf{Q}\mathbf{M} = (\bar{\mathbf{x}}\mathbf{e}^T + \mathbf{Q})\mathbf{M}. \quad (37)$$

Hence we introduce the matrix $\mathbf{Y} = \bar{\mathbf{x}}\mathbf{e}^T + \mathbf{Q}$ and rewrite (34)-(35) as a single equation

$$\dot{\mathbf{Y}} = f(\mathbf{Y}\mathbf{M})\mathbf{M}^{-1} - \mathbf{Y}\mathbf{T}\mathbf{S}(t) \quad (38)$$

in $\mathbf{Y} \in \mathbb{R}^{k \times m}$.

4 Numerical implementation

We now describe our basic numerical algorithms. We start again by looking at the linear problem (15) and its continuous factorization formulation (23)-(24). We continue using the notion ‘‘continuous factorization’’ even though time-stepping methods introduce discontinuous changes in the matrix factors from time-step to time-step. However, these discontinuities vanish as the step-size $\Delta t \rightarrow 0$.

4.1 General linear problems

We now describe a particular discretization method for (23)-(24) which is different from the standard approaches proposed, for example, by Dieci and van Vleck (1995); Bridges and Reich (2001). We will generalize the discretization to nonlinear ensemble formulations in the subsequent section.

Let us assume that

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \Delta t \phi(\mathbf{X}_n; \Delta t) \quad (39)$$

is a numerical method for (15). We wish to find a discretization of (23)-(24) such that

$$\mathbf{X}_n = \mathbf{Q}_n \mathbf{M}_n \quad (40)$$

for all $n \geq 0$ independent of the specific choice of \mathbf{S} . We propose

$$\mathbf{Q}_{n+1} = \mathbf{Q}_n + \Delta t \phi(\mathbf{Q}_n \mathbf{M}_n; \Delta t) \mathbf{M}_{n+1/2}^{-1} - \Delta t \mathbf{Q}_{n+1/2} \mathbf{S}_{n+1/2}, \quad (41)$$

$$\mathbf{M}_{n+1} = \mathbf{M}_n + \Delta t \mathbf{S}_{n+1/2} \mathbf{M}_{n+1/2}. \quad (42)$$

Here we have introduced the standard midpoint approximation $\mathbf{Y}_{n+1/2} = (\mathbf{Y}_n + \mathbf{Y}_{n+1})/2$.

Multiply (41) by $\mathbf{M}_{n+1/2}$ from the right and (42) by $\mathbf{Q}_{n+1/2}$ from the left to obtain

$$\mathbf{Q}_{n+1} \mathbf{M}_{n+1} = \mathbf{Q}_n \mathbf{M}_n + \Delta t \phi(\mathbf{Q}_n \mathbf{M}_n; \Delta t) = \mathbf{X}_{n+1} \quad (43)$$

as a linear combination of the two resulting equations.

Following Bridges and Reich (2001), the symmetric part of $\mathbf{S}_{n+1/2}$ is given by

$$\mathbf{S}_{n+1/2}^{\text{sym}} = \frac{1}{2} \left[\mathbf{Q}_{n+1/2}^T \phi(\mathbf{Q}_n \mathbf{M}_n; \Delta t) \mathbf{M}_{n+1/2}^{-1} + \mathbf{M}_{n+1/2}^{-T} [\phi(\mathbf{Q}_n \mathbf{M}_n; \Delta t)]^T \mathbf{Q}_{n+1/2} \right]. \quad (44)$$

To keep \mathbf{M}_{n+1} symmetric, $\mathbf{S}_{n+1/2}$ needs to satisfy

$$\mathbf{S}_{n+1/2} \mathbf{M}_{n+1/2} - \mathbf{M}_{n+1/2} \mathbf{S}_{n+1/2}^T = \mathbf{0}, \quad (45)$$

which defines the skew-symmetric part $\mathbf{S}_{n+1/2}^{\text{skew}}$.

The numerical approximation \mathbf{Q}_{n+1} , as given by (41), does not exactly satisfy the constraint $\mathbf{Q}_{n+1}^T \mathbf{Q}_{n+1} = \mathbf{I}_m$. As suggested by Higham (1997); Bridges and Reich (2001), we apply a stabilization technique and iterate

$$\mathbf{Q}_{n+1}^{i+1} = \mathbf{Q}_{n+1}^i - \frac{1}{2} \mathbf{Q}_{n+1}^i \left[(\mathbf{Q}_{n+1}^i)^T \mathbf{Q}_{n+1}^i - \mathbf{I}_m \right] \quad (46)$$

for $i = 0, \dots, i_*$ (or to convergence) with $\mathbf{Q}_{n+1}^0 = \mathbf{Q}_{n+1}$. The time-stepping algorithm is then continued with $\mathbf{Q}_{n+1} := \mathbf{Q}_{n+1}^{i_*}$.

4.2 Nonlinear ensemble propagation

Having considered the linear problem in detail, a numerical implementation of the nonlinear problem (34)-(35) is now straightforward. We consider the numerical approximation

$$\mathbf{X}_n = \bar{\mathbf{x}}_n \mathbf{e}^T + \mathbf{Q}_n \mathbf{M}_n = \mathbf{Y}_n \mathbf{M}_n \quad (47)$$

and propose the update

$$\mathbf{Y}_{n+1} = \mathbf{Y}_n + \Delta t \phi(\mathbf{X}_n; \Delta t) \mathbf{M}_{n+1/2}^{-1} - \Delta t \mathbf{Y}_{n+1/2} \mathbf{T} \mathbf{S}_{n+1/2}. \quad (48)$$

The discrete time-evolution of \mathbf{M} is still given by (42) and it is easy to verify that

$$\mathbf{Y}_{n+1} \mathbf{M}_{n+1} = \mathbf{X}_n + \Delta t \phi(\mathbf{X}_n; \Delta t) = \mathbf{X}_{n+1}. \quad (49)$$

The symmetric part of $\mathbf{S}_{n+1/2}$ is defined by

$$\mathbf{S}_{n+1/2}^{\text{sym}} = \frac{1}{2} \left[\mathbf{T} \mathbf{Y}_{n+1/2}^T \phi(\mathbf{X}_n; \Delta t) \mathbf{T} \mathbf{M}_{n+1/2}^{-1} + \mathbf{M}_{n+1/2}^{-T} \mathbf{T} [\phi(\mathbf{X}_n; \Delta t)]^T \mathbf{Y}_{n+1/2} \mathbf{T} \right]. \quad (50)$$

If one additionally wishes to keep \mathbf{M}_{n+1} symmetric, then $\mathbf{S}_{n+1/2}$ needs to satisfy again (45), which defines the skew-symmetric part $\mathbf{S}_{n+1/2}^{\text{skew}}$.

As for the linear case, the numerical approximation $\mathbf{Q}_{n+1} = \mathbf{Y}_{n+1} \mathbf{T}$ does not exactly satisfy the constraint $\mathbf{Q}_{n+1}^T \mathbf{Q}_{n+1} = \mathbf{T}$. Hence we apply a modified stabilization technique and iterate

$$\mathbf{Y}_{n+1}^{i+1} = \mathbf{Y}_{n+1}^i - \frac{1}{2} \mathbf{Y}_{n+1}^i \mathbf{T} \left[(\mathbf{Y}_{n+1}^i \mathbf{T})^T \mathbf{Y}_{n+1}^i \mathbf{T} - \mathbf{T} \right] \quad (51)$$

for $i = 0, \dots, i_*$ (or to convergence) with $\mathbf{Y}_{n+1}^0 = \mathbf{Y}_{n+1}$. The time-stepping algorithm is then continued with $\mathbf{Y}_{n+1} := \mathbf{Y}_{n+1}^{i_*}$.

The time-stepping methods described in this section is implicit in \mathbf{Y}_{n+1} and \mathbf{M}_{n+1} . A simplified implementation based on a single fixed point iteration is described in the Appendix.

4.3 SVD post-processing

We have proposed algorithms that lead to matrix factorizations $\mathbf{X}'_n = \mathbf{Q}_n \mathbf{M}_n$ and $\mathbf{X}_n = \mathbf{Y}_n \mathbf{M}_n$, respectively, at each time level t_n , which become continuous in t as $\Delta t \rightarrow 0$.

We now describe a simple post-processing procedure to find an approximative SVD for \mathbf{M}_{n+1} given $\mathbf{M}_n \approx \mathbf{U}_n \boldsymbol{\Sigma}_n \mathbf{V}_n^T$ and Δt sufficiently small.¹

We first consider the case that \mathbf{M}_{n+1} is symmetric and positive definite (polar factorization), i.e., $\mathbf{U}_n = \mathbf{V}_n$. Let us use the notation \mathbf{R}^{diag} to denote the diagonal part of a matrix \mathbf{R} . We start by defining $\mathbf{R}_0 = \mathbf{V}_n^T \mathbf{M}_{n+1} \mathbf{V}_n$. Then we set $\mathbf{D}_0 = \mathbf{R}_0^{\text{diag}}$ and $\mathbf{W}_0 = \mathbf{V}_n^T$. Note that $\|\mathbf{R}_0 - \mathbf{D}_0\| = \mathcal{O}(\Delta t)$. The following recursion is performed for $i = 0, \dots, i_*$ (or till convergence):

1. Determine the skew-symmetric matrix \mathbf{A}_i from

$$\mathbf{A}_i \mathbf{D}_i + \mathbf{D}_i \mathbf{A}_i^T = \mathbf{R}_i - \mathbf{D}_i. \quad (52)$$

2. Compute

$$\mathbf{L}_i = \left(\mathbf{I}_m - \frac{1}{2} \mathbf{A}_i \right)^{-1} \left(\mathbf{I}_m + \frac{1}{2} \mathbf{A}_i \right). \quad (53)$$

3. Set $\mathbf{W}_{i+1} = \mathbf{L}_i^T \mathbf{W}_i$, $\mathbf{R}_{i+1} = \mathbf{L}_i^T \mathbf{R}_i \mathbf{L}_i$, and $\mathbf{D}_{i+1} = \mathbf{R}_{i+1}^{\text{diag}}$.

We finally set $\mathbf{V}_{n+1} = \mathbf{W}_{i_*}^T$ and $\boldsymbol{\Sigma}_{n+1} = \mathbf{D}_{i_*}$.

We note that the algorithm is subject to the same division by zero problem that other continuous SVD algorithms suffers from whenever two singular values become identical (in which case (52) is not solvable for \mathbf{A}_i).² However, our numerical results suggest (compare Figure 4) that the algorithm provides an approximative SVD for \mathbf{M}_{n+1} provided Δt is sufficiently small.

The SVD post-processing algorithm can be interpreted as a discretization of underlying ODEs in \mathbf{W} and \mathbf{R} . These ODEs are related to equations proposed by Baumann and Helmke (2003). However, in our case, the orthogonality of the transformation matrices \mathbf{W}_i is explicitly built in through the use of the Cayley transform in (53).

In case \mathbf{M}_{n+1} is not symmetric, we assume that an approximative SVD

$$\mathbf{M}_n^T \mathbf{M}_n \approx \mathbf{V}_n \boldsymbol{\Sigma}_n^2 \mathbf{V}_n^T \quad (54)$$

is given to compute an approximative SVD

$$\mathbf{V}_{n+1} \boldsymbol{\Sigma}_{n+1}^2 \mathbf{V}_{n+1}^T \approx \mathbf{M}_{n+1}^T \mathbf{M}_{n+1} \quad (55)$$

for $\mathbf{M}_{n+1}^T \mathbf{M}_{n+1}$ using the algorithm described above. An approximative SVD of \mathbf{M}_{n+1} is then provided by

$$\mathbf{M}_{n+1} \approx \mathbf{U}_{n+1} \boldsymbol{\Sigma}_{n+1} \mathbf{V}_{n+1}^T \quad (56)$$

with

$$\mathbf{U}_{n+1} = \mathbf{M}_{n+1} \mathbf{V}_{n+1} [\boldsymbol{\Sigma}_{n+1}]^{-1}. \quad (57)$$

Note that \mathbf{U}_{n+1} , as defined in (57), is only almost orthogonal, in general.

¹One could also consider the SVD of $\mathbf{M}_n - \mathbf{w} \mathbf{e}^T$. Both SVDs differ only in the singular value associated with the common singular vector \mathbf{w} , i.e., one has $\mathbf{M}_n \mathbf{w} = \mathbf{w}$ and $(\mathbf{M}_n - \mathbf{w} \mathbf{e}^T) \mathbf{w} = \mathbf{0}$.

²If two diagonal entries d_p and d_q , $q \neq p$, of \mathbf{D}_i become identical or nearly identical, then we set the corresponding entries of \mathbf{A}_i equal to zero, i.e., $a_{qp} = a_{pq} = 0$, in our implementation of the algorithm.

5 Applications

We now apply the continuous factorization algorithm to the Lorenz-96 model (Lorenz, 1996; Lorenz and Emanuel, 1998). The particular implementation of the algorithm is described in the Appendix.

The standard implementation of the Lorenz-96 model has state vector $\mathbf{x} = (x_1, \dots, x_k)^T \in \mathbb{R}^k$, $k = 40$, and its time evolution is given by the differential equations

$$\dot{x}_j = (x_{j+1} - x_{j-2})x_{j-1} - x_j + 8 \quad (58)$$

for $j = 1, \dots, k$. To close the equations, we define $x_{-1} = x_{39}$, $x_0 = x_{40}$, and $x_{41} = x_1$.

The Lorenz-96 equations are solved numerically using the implicit midpoint rule with $\Delta t = 0.005$. All continuous factorizations are performed using the algorithm described in the Appendix.

5.1 Ensemble breeding for Lorenz-96 model

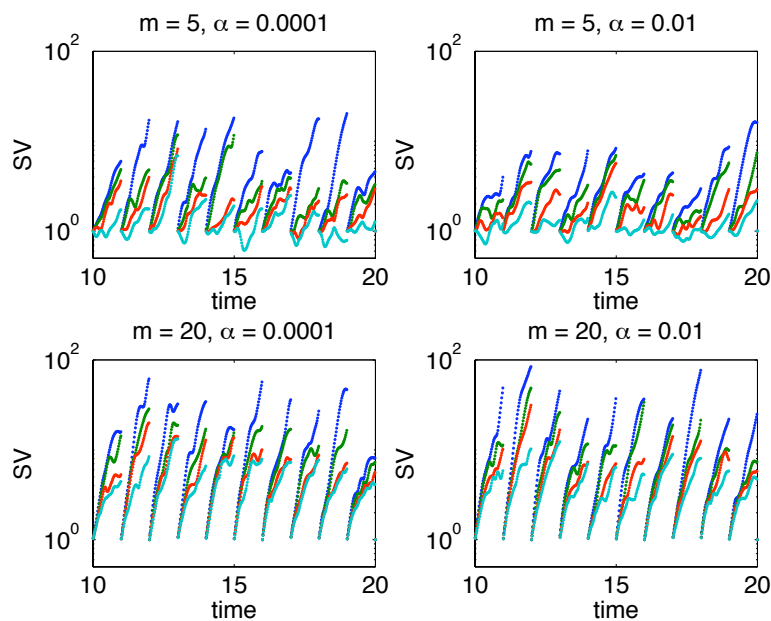


Figure 1: Ensemble propagation over several “breeding” cycles of length $\Delta t_{\text{breed}} = 1.0$ for two different ensemble sizes ($m = 5$ and $m = 20$), and two different values of the scaling parameter α ($\alpha = 0.01$ and $\alpha = 0.0001$). Displayed are the four largest singular values of \mathbf{M}_n/α from a (continuous) SVD of \mathbf{X}'_n at $t_n = n \cdot \Delta t \in [10, 20]$ with $\Delta t = 0.005$. Colors cyan, red, dark green, and blue are used to denote the singular values in increasing magnitude at any given instance in time.

We compute a continuous SVD factorization for periodically reset ensembles $\mathbf{X}_n = \mathbf{Y}_n \mathbf{M}_n$, which amounts to a particular implementation of the ensemble breeding method (Toth and Kalnay, 1993, 1997).

Assume an ensemble resetting is performed at t_{breed} . We introduce the notations $\mathbf{M}(t_{\text{breed}} - \varepsilon)$ and $\mathbf{M}(t_{\text{breed}} + \varepsilon)$ to denote the value of $\mathbf{M}(t)$ before and after the resetting step, i.e., we consider the limit $\varepsilon \rightarrow 0$ and note that $\mathbf{M}(t)$ is discontinuous across t_{breed} . Resetting is performed in our experiments by simply replacing $\mathbf{M}(t_{\text{breed}} - \varepsilon)$ by

$$\mathbf{M}(t_{\text{breed}} + \varepsilon) = \alpha \mathbf{T} + \mathbf{w} \mathbf{e}^T, \quad (59)$$

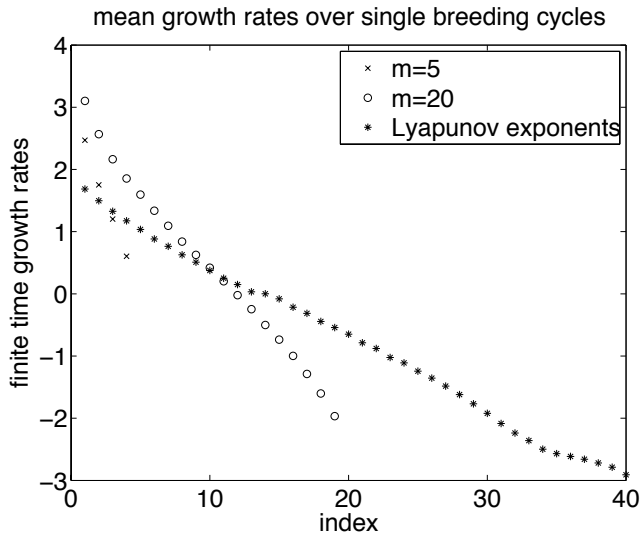


Figure 2: We display the finite-time growth rates averaged over all breeding cycles using two different ensemble sizes of $m = 5$ and $m = 20$. Finite-time growth rates are computed over a single breeding cycle using (61). Note that the smaller ensemble leads to significantly smaller local growth rates. We also plot the standard Lyapunov exponents for the Lorenz-96 model for comparison.

where $\alpha > 0$ is an appropriate scaling factor. The matrix $\mathbf{Y}(t_{\text{breed}})$ is not changed during the resetting step, which implies that the ensemble mean and the image of \mathbf{X}'_n are preserved under the resetting. Note that other ensemble resettings could be performed. See, for example, Toth and Kalnay (1993, 1997); Wei et al. (2008). The ensemble resetting is performed every Δt_{breed} time units.

In Fig. 1 we display the time evolution of the four dominant singular values of \mathbf{M}_n/α over several breeding cycles for $\Delta t_{\text{breed}} = 1.0$, two different ensemble sizes ($m = 5$, and $m = 20$) and two different values of α ($\alpha = 0.0001$, $\alpha = 0.01$). We conclude that the larger ensemble ($m = 20$) leads to a faster growth in the singular values. To confirm this observation, we also compute the finite time-growth rates over each breeding cycle. The finite-time growth rates are determined by the singular values of

$$\mathbf{M}(t_{\text{breed}} - \varepsilon)[\mathbf{M}(t_{\text{breed}} + \Delta t_{\text{breed}} + \varepsilon)]^{-1} = \mathbf{M}(t_{\text{breed}} - \varepsilon)[\alpha^{-1}\mathbf{T} + \mathbf{w}\mathbf{e}^T]. \quad (60)$$

Hence, if σ_i , $i = 1, \dots, m$, are the singular values of $\mathbf{M}(t_{\text{breed}} - \varepsilon)$ and if we assume that $\sigma_1 = 1$ is the singular value corresponding to the singular vector \mathbf{w} , then the singular values of (60) are given by $\sigma_1 = 1$ and σ_i/α , $i = 2, \dots, m$. The finite-time growth rates are now defined by

$$\lambda_i(t_{\text{breed}}) = \frac{1}{\Delta t_{\text{breed}}} \log \left(\frac{\sigma_{i+1}}{\alpha} \right), \quad i = 1, \dots, m - 1. \quad (61)$$

The averaged values

$$\bar{\lambda}_i = \frac{1}{J} \sum_{j=1}^J \lambda_i(j \cdot \Delta t_{\text{breed}}), \quad i = 1, \dots, m - 1, \quad (62)$$

over $J = 100$ breeding cycles are displayed in Fig. 2 for $\Delta t_{\text{breed}} = 1.0$, $\alpha = 0.01$, and ensembles of size $m = 20$ and $m = 5$. The smaller ensemble leads to a reduced finite-time growth rate confirming the results from Figure 1. We verified numerically that our results are in agreement with standard finite-time Lyapunov exponents (Goldhirsch et al., 1987). For comparison, we also computed the standard Lyapunov exponents for the Lorenz-96 model using a discrete QR algorithm (Geist et al., 1990).

5.2 Ensemble Kalman filter for Lorenz-96 model

We now combine ensemble prediction with a Kalman analysis step, which forms the core of the ensemble Kalman filter (EnKF) approach (Evensen, 2006). Let us assume that observations are available at time t_{obs} . We first summarize the Kalman analysis step.

5.2.1 Kalman analysis step

Given a forecast covariance matrix

$$\mathbf{P}_f = \frac{1}{m-1} \mathbf{X}'(t_{\text{obs}} - \varepsilon) [\mathbf{X}'(t_{\text{obs}} - \varepsilon)]^T = \frac{1}{m-1} \mathbf{X}(t_{\text{obs}} - \varepsilon) \mathbf{T} [\mathbf{X}(t_{\text{obs}} - \varepsilon)]^T, \quad (63)$$

a forecast mean

$$\mathbf{x}_f = \mathbf{X}(t_{\text{obs}} - \varepsilon) \mathbf{w}, \quad (64)$$

and observations

$$\mathbf{y}(t_{\text{obs}}) = \mathbf{H} \mathbf{x}_f + \mathbf{r}(t_{\text{obs}}) \in \mathbb{R}^l, \quad (65)$$

where $\mathbf{r}(t_{\text{obs}}) \in \mathbb{R}^l$ are i.i.d. Gaussian random numbers with mean zero and covariance matrix $\mathbf{R} \in \mathbb{R}^{l \times l}$, the Kalman analysis step first computes the Kalman gain matrix

$$\mathbf{K} = \mathbf{P}_f \mathbf{H}^T (\mathbf{H} \mathbf{P}_f \mathbf{H}^T + \mathbf{R})^{-1} \quad (66)$$

and then updates the forecast mean to

$$\mathbf{x}_a = \mathbf{x}_f - \mathbf{K}(\mathbf{H} \mathbf{x}_f - \mathbf{y}(t_{\text{obs}})) \quad (67)$$

and the forecast covariance matrix to

$$\mathbf{P}_a = \mathbf{P}_f - \mathbf{K} \mathbf{H} \mathbf{P}_f. \quad (68)$$

The ensemble mean is now changed to

$$\bar{\mathbf{x}}(t_{\text{obs}} + \varepsilon) = \mathbf{X}(t_{\text{obs}} + \varepsilon) \mathbf{w} = \mathbf{x}_a. \quad (69)$$

The update of the ensemble deviation matrix \mathbf{X}' is more delicate and we use the following approach.

Given the updated covariance matrix \mathbf{P}_a , the updated ensemble deviation matrix \mathbf{X}'_a needs to satisfy

$$\mathbf{P}_a = \frac{1}{m-1} \mathbf{X}'_a [\mathbf{X}'_a]^T. \quad (70)$$

The desired update of the ensemble deviation matrix can be achieved in many different ways. Ensemble transform Kalman filters (ETKFs) (a variant of ensemble square root filters) offer a particularly elegant approach (Bishop et al., 2001; Tippett et al., 2003; Wang et al., 2004). ETKFs are based on a transformation matrix $\mathbf{F} \in \mathbb{R}^{m \times m}$ such that

$$\mathbf{X}'_a = \mathbf{X}'(t_{\text{obs}} - \varepsilon) \mathbf{F} \quad (71)$$

It is also desirable that $\mathbf{F} \mathbf{w} = \mathbf{0}$, which implies $\mathbf{X}'_a \mathbf{w} = \mathbf{0}$, i.e., the covariance update is unbiased (Wang et al., 2004; Livings et al., 2008).

Here we suggest determining \mathbf{X}'_a as the solution of the continuous Kalman filter equation (Simon, 2006)

$$\frac{d}{ds} \tilde{\mathbf{X}}' = -\frac{1}{2m-2} \tilde{\mathbf{X}}' [\tilde{\mathbf{X}}']^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \tilde{\mathbf{X}}' \quad (72)$$

at $s = 1$, i.e., $\mathbf{X}'_a = \tilde{\mathbf{X}}'(1)$, subject to the initial condition $\tilde{\mathbf{X}}(0) = \mathbf{X}'(t_{\text{obs}} - \varepsilon)$. It follows from standard results (Simon, 2006) that $\mathbf{X}'_a = \tilde{\mathbf{X}}'(1)$ satisfies (70) and that $\mathbf{X}'_a \mathbf{w} = \mathbf{0}$. We finally set

$$\mathbf{X}'(t_{\text{obs}} + \varepsilon) = \mathbf{X}'_a \quad (73)$$

and a complete update of the ensemble is obtained.

In fact, the solution $\tilde{\mathbf{X}}'(1)$ of (72) can be stated in the form (71) and the transformation matrix \mathbf{F} is identical to the transformation proposed in eq. (C1) of Wang et al. (2004). However, in our implementations, we replace the explicit update (71) by a numerical integration of (72) with the continuous factorization algorithm. This approach allows us to obtain a (continuously embedded) factorization $\mathbf{X}'_a = \mathbf{Q}_a \mathbf{M}_a$, i.e.

$$\mathbf{X}'(t_{\text{obs}} + \varepsilon) = \mathbf{Q}(t_{\text{obs}} + \varepsilon) \mathbf{M}(t_{\text{obs}} + \varepsilon) \quad (74)$$

with

$$\mathbf{Q}(t_{\text{obs}} + \varepsilon) = \mathbf{Q}_a, \quad \mathbf{M}(t_{\text{obs}} + \varepsilon) = \mathbf{M}_a. \quad (75)$$

We also note that, alternatively, one could pre-multiply the ensemble deviation matrix $\mathbf{X}'(t_{\text{obs}} - \varepsilon)$ by an appropriate matrix $\mathbf{A} \in \mathbb{R}^{k \times k}$ such that

$$\mathbf{X}'_a = \mathbf{A} \mathbf{X}'(t_{\text{obs}} - \varepsilon) \quad (76)$$

as proposed by Anderson (2001). The associated ensemble adjustment Kalman filter obviously satisfies $\mathbf{X}'_a \mathbf{w} = \mathbf{0}$. The matrix \mathbf{A} can also be obtained from (72) but, since we are mainly interested in the case $k \gg m$, the representation (71) is preferred.

Computational savings are possible by introducing the variable

$$\mathbf{Z}(s) = \mathbf{Q}^T(t_{\text{obs}} - \varepsilon) \tilde{\mathbf{X}}'(s) \in \mathbb{R}^{m \times m} \quad (77)$$

which transforms (72) into the differential equation

$$\frac{d}{ds} \mathbf{Z} = -\frac{1}{2m-2} \mathbf{Z} \mathbf{Z}^T \mathbf{C} \mathbf{Z} \quad (78)$$

in \mathbf{Z} with a constant and symmetric matrix

$$\mathbf{C} = \mathbf{Q}^T(t_{\text{obs}} - \varepsilon) \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{Q}(t_{\text{obs}} - \varepsilon). \quad (79)$$

The initial conditions are given by

$$\mathbf{Z}(0) = \mathbf{Q}^T(t_{\text{obs}} - \varepsilon) \tilde{\mathbf{X}}'(0) = \mathbf{Q}^T(t_{\text{obs}} - \varepsilon) \mathbf{Q}(t_{\text{obs}} - \varepsilon) \mathbf{M}(t_{\text{obs}} - \varepsilon) = \mathbf{T} \mathbf{M}(t_{\text{obs}} - \varepsilon). \quad (80)$$

A factorization $\mathbf{Z}(s) = \mathbf{U}(s) \tilde{\mathbf{M}}(s)$ can now be computed using techniques from Section 4. The initial conditions are $\mathbf{U}(0) = \mathbf{T}$ and $\tilde{\mathbf{M}}(0) = \mathbf{M}(t_{\text{obs}} - \varepsilon)$.

The updated ensemble factorization is given by

$$\mathbf{Q}(t_{\text{obs}} + \varepsilon) = \mathbf{Q}(t_{\text{obs}} - \varepsilon) \mathbf{U}(1), \quad \mathbf{M}(t_{\text{obs}} + \varepsilon) = \tilde{\mathbf{M}}(1). \quad (81)$$

5.2.2 Continuous matrix factorization EnKF

A single step of the continuous matrix factorization EnKF consists of the following three sub-steps:

1. Given an analyzed ensemble $\mathbf{X}(t_{\text{obs}} - \Delta t_{\text{obs}} + \varepsilon)$ at previous observation time $t_{\text{obs}} - \Delta t_{\text{obs}}$ and its polar decomposition, apply the continuous matrix factorization algorithms from the Appendix to the ensemble dynamics (7) over a time-interval of length Δt_{obs} with step-size Δt . We obtain the forecast ensemble $\mathbf{X}(t_{\text{obs}} - \varepsilon)$ and its polar decomposition at observation time t_{obs} .

- If necessary, an ensemble inflation (Anderson and Anderson, 1999) is performed prior to the Kalman analysis step by replacing $\mathbf{M}(t_{\text{obs}} - \varepsilon)$ with

$$\delta [\mathbf{M}(t_{\text{obs}} - \varepsilon) - \mathbf{w}\mathbf{e}^T] + \mathbf{w}\mathbf{e}^T, \quad (82)$$

where $\delta > 1$ is appropriately chosen. Note that the “inflated” \mathbf{M} still satisfies $\mathbf{M}\mathbf{w} = \mathbf{w}$.

- Apply the continuous matrix factorization algorithms from the Appendix to the differential equation (78) with initial condition $\mathbf{Z}(0) = \mathbf{T}\mathbf{M}(t_{\text{obs}} - \varepsilon)$ over a unit time-interval with step-size Δs . Denote the numerical result at $s = 1$ by $\mathbf{Z}(1) = \mathbf{U}(1)\tilde{\mathbf{M}}(1)$. The analyzed ensemble deviation matrix at t_{obs} and its polar factorization are given by

$$\mathbf{X}'(t_{\text{obs}} + \varepsilon) = \mathbf{Q}(t_{\text{obs}} + \varepsilon)\mathbf{M}(t_{\text{obs}} + \varepsilon) = \mathbf{Q}(t_{\text{obs}} - \varepsilon)\mathbf{U}(1)\tilde{\mathbf{M}}(1), \quad (83)$$

i.e., $\mathbf{Q}(t_{\text{obs}} + \varepsilon) = \mathbf{Q}(t_{\text{obs}} - \varepsilon)\mathbf{U}(1)$ and $\mathbf{M}(t_{\text{obs}} + \varepsilon) = \tilde{\mathbf{M}}(1)$. The ensemble mean is updated according to (69).

If desired, compute an approximative SVD of \mathbf{M} and $\tilde{\mathbf{M}}$, respectively, after each update in Items 1-3 above.

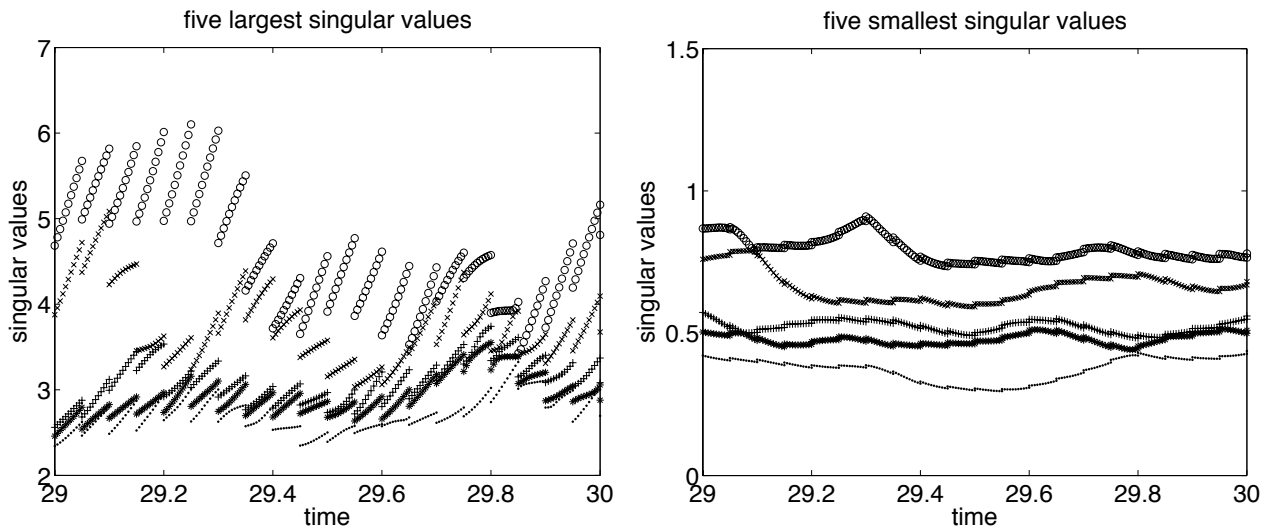


Figure 3: Five largest (left) and five smallest (right) singular values of \mathbf{M}_n over a time-interval $t_n \in [29, 30]$ and for $m = 20$ ensemble members. One can clearly see that the analysis step strongly reduces the largest singular values, while it can exert a mildly increasing effect on small singular values. In both panels, the symbols \cdot , $*$, $+$, \times , and \circ are used to denote the singular values in increasing magnitude at any given instance in time.

5.2.3 Numerical results

We have implemented the algorithm from Section 5.2.2 and the unbiased ensemble transform Kalman filter (ETKF) of Wang et al. (2004) for the Lorenz-96 model. Our implementations use $\Delta t_{\text{obs}} = 0.05$, $\Delta t = 0.005$, $\Delta s = 0.5$ (i.e., two integration steps for (78) per analysis cycle), $l = 20$ (i.e., every second grid point is observed), and measurement error covariance matrix $\mathbf{R} = \mathbf{I}_l$. We found numerically that $\Delta s = 1.0$ leads to filter divergence while $\Delta s < 0.5$ does not significantly change the results. A total of 1000 analysis steps are performed in each experiment. The “true” trajectory $\mathbf{x}_{\text{truth}}(t_n)$ is generated by integrating the Lorenz-96 model

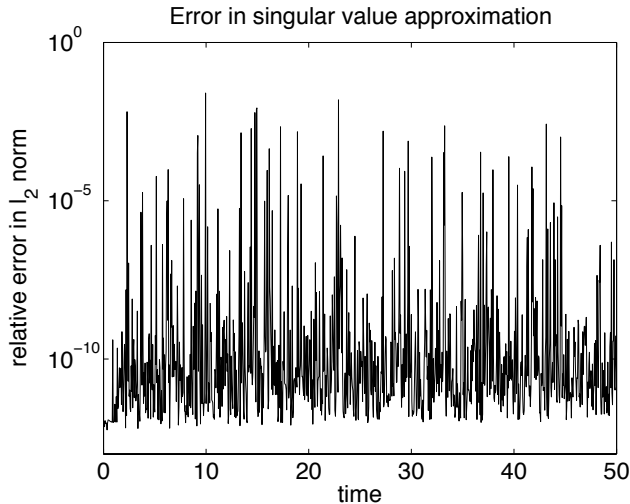


Figure 4: We compare the exact singular values of \mathbf{M}_n (as computed with MATLAB) with their approximation obtained from the SVD post-processing algorithm as described in Section 4.3. We performed two iterations of the SVD post-processing algorithm after each time-step, i.e., we set $i_* = 2$. Displayed is the l_∞ -norm of the relative difference in the singular values. We conclude that SVD post-processing is a very cheap and robust method for "continuously" updating singular values.

with the implicit midpoint rule and step-size $\Delta t = 0.005$, i.e., we assume that there is no model error. The observations are obtained according to

$$\mathbf{y}(t_{\text{obs}}) = \mathbf{H}\mathbf{x}_{\text{truth}}(t_{\text{obs}}) + \mathbf{r}(t_{\text{obs}}), \quad (84)$$

where $\mathbf{r}(t_{\text{obs}})$ are i.i.d. Gaussian random numbers with mean zero and covariance matrix \mathbf{R} .

ETKF/factorized EnKF	$m = 17$	$m = 20$	$m = 25$
$\delta = \sqrt{1.05}$	NA	0.2990/0.3074	0.2916/0.2997
$\delta = \sqrt{1.10}$	0.3681/0.3866	0.3260/0.3557	0.3158/0.3330

Table 1: We display the RMS errors from the continuous matrix factorization EnKF of Section 5.2.2 and an implementation of the ETKF of Wang et al. (2004) for different ensemble sizes and inflation factors. We see that the ETKF performs slightly better. The difference is due to finite step-size effects in Δt and Δs . Both Kalman filters are unable to track the reference solution for $m = 17$ and $\delta = \sqrt{1.05}$.

In Table 1, we display the RMS errors

$$\text{rms} = \sqrt{\frac{1}{Jl} \sum_{j=1}^J \|\mathbf{H}[\bar{\mathbf{x}}(j \cdot \Delta t_{\text{obs}}) - \mathbf{x}_{\text{truth}}(j \cdot \Delta t_{\text{obs}})]\|^2} \quad (85)$$

between the truth $\mathbf{x}_{\text{truth}}$ and the ensemble mean $\bar{\mathbf{x}}$ for different values of the ensemble size m and the ensemble inflation factor δ . It can be seen that both filter implementations behave qualitatively the same with the ETKF performing slightly better. The difference in the numerical results is due to finite step-size effects in Δt and Δs and would vanish in the limit $\Delta t, \Delta s \rightarrow 0$.

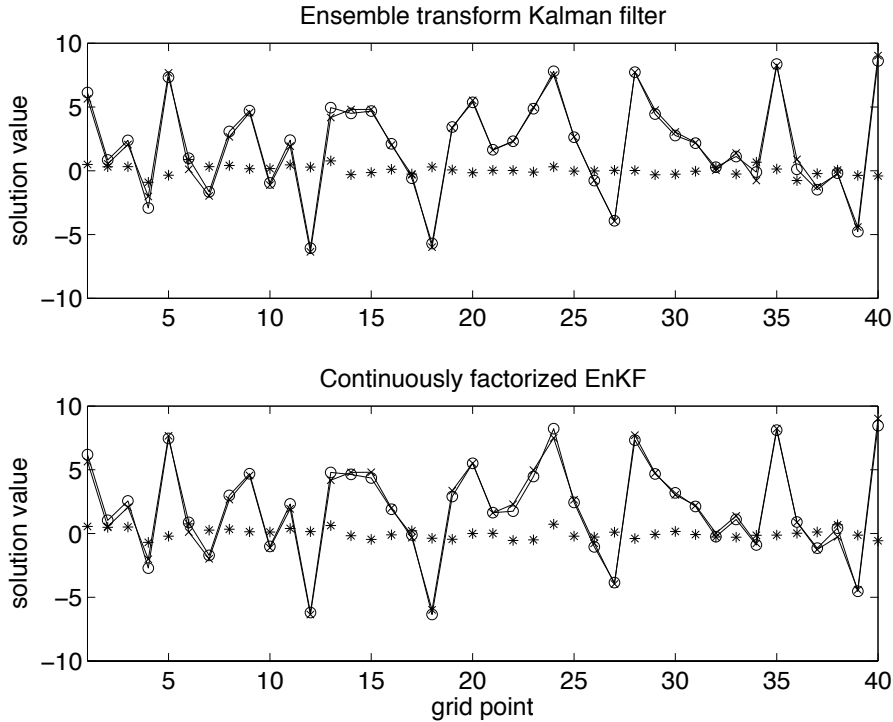


Figure 5: Numerical computed ensemble mean (solid line with circles), its reference value (solid line with crosses), and difference (stars) after 1000 analysis steps for $m = 20$ ensemble members. Both implementations display an essentially identical behavior in terms of mean square root errors.

More specific results for $m = 20$ and $\delta = \sqrt{1.05}$ can be found in Figures 3, 4, and 5, where we show the time evolution of singular values over a few propagation/analysis cycles, the error in the approximative SVD factorization, and the difference in the final ensemble mean $\bar{\mathbf{x}}$ and reference solution $\mathbf{x}_{\text{truth}}$ for both filter implementations. In Fig. 6, we furthermore considered the singular values of $\mathbf{M}(j \cdot \Delta t_{\text{obs}} - \varepsilon)$ for $j = 1, \dots, 1000$ and computed their mean and variance for $m = 20$ and $\delta = \sqrt{1.05}$. It can be concluded that the singular values of \mathbf{P}_f stay close to their mean value and that the mean values form a rather regular curve. Further investigations reveal that the singular vectors, on the other hand, undergo rapid transitions. This observation is, of course, not generic for chaotic dynamical systems.

We finally comment on the computational cost added through the continuous matrix factorization formulations. Using the commands `tic/toc` under MATLAB, we found that a single time-step of the continuous factorization algorithm, as described in the Appendix, takes about 1.6 to 1.7 times as long as a single time-step without factorization for the Lorenz-96 model. More specifically, the average CPU time for performing (95) for an ensemble size of $m = 20$ was found to be 0.0035 seconds while the subsequent factorization step required 0.0024 seconds. The corresponding values for $m = 25$ and $m = 17$ are 0.0050/0.0031 and 0.0029/0.0021 respectively. For more complex models, the CPU time of performing (95) is likely to increase much more rapidly than the CPU time required for the additional algebraic manipulations and the performance ratio between ETKF and factorized EnKF should reduce closer to one. See also the Appendix for a brief discussion on floating point operations.

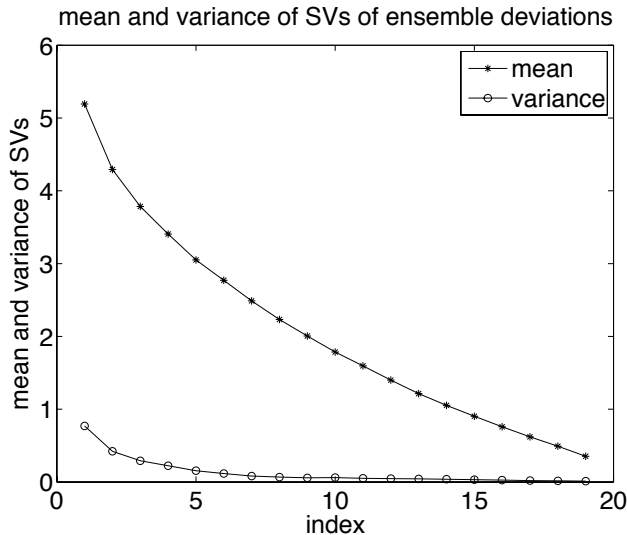


Figure 6: We display the mean and the variance of the singular values of $\mathbf{M}(j \cdot \Delta t_{\text{obs}} - \varepsilon)$ for $m = 20$ and $\delta = \sqrt{1.05}$ when viewed as a time series in $j = 1, \dots, 1000$. We conclude that the singular values stay relatively close to their mean values along the complete simulation cycle.

6 Ensemble re-orthogonalization

So far we have discussed algorithms that provide a matrix factorization $\mathbf{Q}_n \mathbf{M}_n$ for computed ensemble deviation matrices \mathbf{X}'_n and have used matrix factorizations to *analyze* ensemble breeding and ensemble Kalman filters. In this section, we will instead use matrix factorizations to actually *modify* the propagation of ensemble deviation matrices. The basic idea is to "re-orthogonalize" the ensemble deviation matrix after each ensemble update under an EnKF.

Let us first explain the basic idea for a linear ODE (15). We set $\mathbf{X}' = \mathbf{X}$ and do not require that $\mathbf{X}' \mathbf{w} = \mathbf{0}$ for a moment, i.e., $\mathbf{X} \in \mathbb{R}^{k \times m}$, $k > m$, has rank m . We also assume that we got a polar factorization of \mathbf{X} at time t_n , i.e., $\mathbf{X}_n = \mathbf{Q}_n \mathbf{M}_n$ and a SVD of \mathbf{M}_n , i.e., $\mathbf{M}_n = \mathbf{V}_n \boldsymbol{\Sigma}_n \mathbf{V}_n^T$. Now we apply the time integration step (39), (41)- (42) *not* to \mathbf{X}_n but to $\widehat{\mathbf{X}}_n = \widehat{\mathbf{Q}}_n \widehat{\mathbf{M}}_n$ instead, where

$$\widehat{\mathbf{Q}}_n = \mathbf{Q}_n \mathbf{V}_n, \quad \widehat{\mathbf{M}}_n = \boldsymbol{\Sigma}_n. \quad (86)$$

We note that the modification conserves the covariance matrix \mathbf{P}_n , i.e.,

$$\mathbf{P}_n = \frac{1}{m-1} \mathbf{X}_n \mathbf{X}_n^T = \frac{1}{m-1} \widehat{\mathbf{X}}_n \widehat{\mathbf{X}}_n^T. \quad (87)$$

Since

$$\widehat{\mathbf{X}}_n^T \mathbf{P}_n^+ \widehat{\mathbf{X}}_n = (m-1) \mathbf{I}_m \quad (88)$$

we call the proposed modification a re-orthogonalization with respect to the covariance matrix \mathbf{P}_n . Here \mathbf{P}_n^+ denotes the pseudo-inverse of \mathbf{P}_n (Golub and Loan, 1996).

We now generalize this approach to nonlinear ensemble propagation. Let \mathbf{X}_n be the computed ensemble at time-level t_n . Let us assume that a polar factorization $\mathbf{X}_n = \mathbf{Y}_n \mathbf{M}_n$ is available as well as a SVD of $\mathbf{M}_n - \mathbf{w} \mathbf{e}^T$, i.e.

$$\mathbf{M}_n - \mathbf{w} \mathbf{e}^T = \bar{\mathbf{V}}_n \bar{\boldsymbol{\Sigma}}_n \bar{\mathbf{V}}_n^T. \quad (89)$$

Such a SVD can be computed continuously using the algorithm of Section 4.3. Finally, let $\mathbf{V}_T \boldsymbol{\Sigma}_T \mathbf{V}_T^T$ denote the SVD of \mathbf{T} . Note that \mathbf{T} has $m-1$ singular values equal to one and a single zero singular value corresponding to the singular vector \mathbf{w} .

We propose to continue the next time-step with a modified ensemble $\widehat{\mathbf{X}}_n = \widehat{\mathbf{Y}}_n \widehat{\mathbf{M}}_n$ defined by

$$\widehat{\mathbf{M}}_n = \mathbf{V}_T \bar{\Sigma}_n \mathbf{V}_T^T + \mathbf{w} \mathbf{e}^T. \quad (90)$$

and

$$\widehat{\mathbf{Y}}_n = \mathbf{Y}_n \bar{\mathbf{V}}_n \mathbf{V}_T^T. \quad (91)$$

The matrix $\bar{\mathbf{V}}_n \mathbf{V}_T^T$ is uniquely defined up to a multiplicative factor of ± 1 and we always pick this factor such that $\bar{\mathbf{V}}_n \mathbf{V}_T^T \mathbf{w} = \mathbf{w}$ and we obtain

$$\widehat{\mathbf{Y}}_n \mathbf{w} = \mathbf{Y}_n \mathbf{w} = \bar{\mathbf{x}}_n. \quad (92)$$

Since also $\widehat{\mathbf{M}}_n \mathbf{w} = \mathbf{w}$, it follows that $\widehat{\mathbf{X}}_n$ and \mathbf{X}_n have a common mean. One also easily verifies that the induced covariance matrix satisfies

$$\mathbf{P}_n = \frac{1}{m-1} \mathbf{X}'_n [\mathbf{X}'_n]^T = \frac{1}{m-1} \widehat{\mathbf{X}}'_n [\widehat{\mathbf{X}}'_n]^T. \quad (93)$$

We note that $\widehat{\mathbf{X}}'_n$ is, unlike (86), not orthogonal with respect to \mathbf{P}_n in the standard sense, i.e.,

$$[\widehat{\mathbf{X}}'_n]^T \mathbf{P}_n^+ \widehat{\mathbf{X}}'_n = (m-1) \mathbf{T} \quad (94)$$

Since \mathbf{T} is a projector onto the linear sub-space $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^k : \mathbf{x}^T \mathbf{w} = 0\}$, we conclude that $\widehat{\mathbf{X}}'_n$ is orthogonal with respect to \mathbf{P}_n over the linear sub-space \mathcal{S} . See also Wei et al. (2008) for a related discussion.

The modified ensemble propagation algorithm is applied to the EnKF filter described in Section 5.2.2. More precisely, the re-orthogonalization is applied after each update of the ensemble under the continuous matrix factorization EnKF. In all other details the algorithm remains identical to the one described in Section 5.2.2.

Numerical experiments reveal that the re-orthogonalization is most important for small ensemble sizes. We state results for $m = 17$ ensemble members, 2000 analysis steps, and different values of the inflation factor δ in Table 2. Note that the Lorenz-96 model with $k = 40$ and $F = 8$ has 13 positive Lyapunov exponents, which enforces a lower limit on the achievable ensemble size. In this context it is quite remarkable that the re-orthogonalized EnKF was able to track the reference solution over 2000 analysis cycles even for $m = 16$ ensemble members and inflation factor $\delta = \sqrt{1.15}$ (rms 0.3513). None of our ETKF implementations was found to be stable for $m \leq 16$.

In Fig. 7 we display the mean and variance of the singular values of the forecast ensembles. We applied the same parameter setting ($m = 20$, $\delta = \sqrt{1.05}$) as used in Fig. 6 for the EnKF without re-orthogonalization. There is only a small difference between the two results, which indicates that the two algorithms differ mainly in their \mathbf{Q}_n matrices (whose orthogonal columns span the $m - 1$ dimensional “ensemble deviations” sub-space of \mathbb{R}^k). This difference must be responsible for the improved performance of the re-orthogonalized EnKF. We also measured the CPU time added to the continuous factorization algorithm through the ensemble re-orthogonalization step and found a value of 0.0009 seconds for $m = 20$.

We conclude this section by mentioning two alternative modifications to ensemble propagation. Localization of the forecast covariance matrix in the Kalman analysis step has been proposed by Houtekamer and Mitchell (2001) and has become a very popular method to avoid spurious correlations associated with remote observations. At present, we cannot see a direct link between localization and re-orthogonalization as proposed here. However, it appears feasible to build localization directly into the ODE (72) and to derive a “localized” continuous matrix factorization EnKF. A much more radical modification to ensemble propagation has been proposed by Harlim and Majda (2008), where nonlinear ensemble propagation is replaced by appropriately fitted linear stochastic differential equations.

	$\delta = \sqrt{1.08}$	$\delta = \sqrt{1.09}$	$\delta = \sqrt{1.10}$	$\delta = \sqrt{1.11}$
ETKF	NA	NA	0.3700	0.6005
re-orthogonalized & factorized EnKF	0.3060	0.3154	0.3212	0.3295

Table 2: We display the RMS errors from the re-orthogonalized EnKF and an implementation of the ensemble transform Kalman filter (ETKF) of Wang et al. (2004) for $m = 17$, 2000 analysis steps, and different values of the ensemble inflation factor δ . NA means that the filter was unable to track the reference solution. We see that the re-orthogonalization improves the filter performance in terms of stability and accuracy.

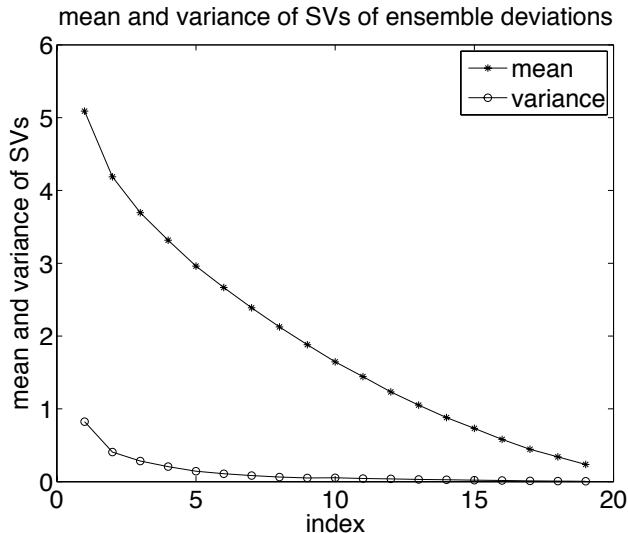


Figure 7: Displayed are the same quantities as in Fig. 6 for the EnKF with re-orthogonalization. It can be concluded that the re-orthogonalization has a rather small impact on the singular values compared to ensemble inflation techniques. In particular, the mean values are slightly smaller for the re-orthogonalized EnKF. More specifically, an increase of the inflation factor from $\delta = \sqrt{1.05}$ to $\delta = \sqrt{1.10}$ leads to an increase in the mean of the largest singular value by about 0.8 while re-orthogonalization reduces the mean of the largest singular value by about 0.1.

7 Conclusions

We have extended and applied continuous matrix factorization algorithms to nonlinear ensemble propagation. Such factorization can be used to analyze the ensemble "continuously" for little additional computational cost. The key idea is that continuous matrix factorizations lead to incremental updates which can be implemented efficiently on machines that execute matrix-matrix multiplications at high efficiency (Golub and Loan, 1996). Continuous matrix factorizations have been used before for computing Lyapunov exponents (Geist et al., 1990; Dieci and van Vleck, 1995; Bridges and Reich, 2001). Here we have demonstrated by means of the Lorenz-96 model that such factorizations can also be implemented for nonlinear ensemble breeding and ensemble Kalman filtering. We have also shown how the matrix factorization can be used to perform a re-orthogonalization of the ensemble after each time step of an ensemble propagation algorithm. The re-orthogonalization introduces a small but non-trivial change to nonlinear ensemble propagation. Numerical evidence from the Lorenz-96 model indicates that this modification improves the behavior of an EnKF.

Acknowledgment. We would like to thank Andreas Hense for discussions on his work on orthogonalized bred vectors, which stimulated us to re-visit our work on continuous polar and QR factorizations (Bridges and Reich, 2001) in the context of nonlinear ensemble propagation.

References

- J.L. Anderson. An ensemble adjustment filter for data assimilation. *Mon. Wea. Rev.*, 129:2884–2903, 2001.
- J.L. Anderson and S.L. Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Mon. Wea. Rev.*, 127:2741–2758, 1999.
- M. Baumann and U. Helmke. Singular value decomposition of time-varying matrices. *Future Generation Computer Systems*, 19:353–361, 2003.
- C.H. Bishop, B. Etherton, and S.J. Majumdar. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Mon. Wea. Rev.*, 129:420–436, 2001.
- Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Comm. Pure Appl. Math.*, 44:375–417, 1991.
- Th.J. Bridges and S. Reich. Computing Lyapunov exponents on a Stiefel manifold. *Physica D*, 156:219–238, 2001.
- L. Dieci and C. Elia. The singular value decomposition to approximate spectra of dynamical systems. Theoretical aspects. *J. Diff. Equ.*, 230:502–531, 2006.
- L. Dieci and E.S. van Vleck. Computation of a few Lyapunov exponents for continuous and discrete dynamical systems. *Appl. Num. Math.*, 17:275–291, 1995.
- G. Evensen. *Data assimilation, the ensemble Kalman filter*. Springer-Verlag, New York, 2006.
- K. Geist, U. Parlitz, and W. Lauterborn. Comparison of different methods for computing Lyapunov exponents. *Prog. Theor. Phys.*, 83:875–893, 1990.
- I. Goldhirsch, P.-L. Sulem, and S.A. Orszag. Stability and Lyapunov stability of dynamical systems: A differential approach and a numerical method. *Physica D*, 27:311–337, 1987.
- G.H. Golub and Ch.F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.
- J.M. Greene and J.S. Kim. The calculation of Lyapunov spectra. *Physica D*, 24:213–225, 1987.
- J. Harlim and A. Majda. Filtering nonlinear dynamical systems with linear stochastic models. *Nonlinearity*, 21:1281–1306, 2008.
- D. Higham. Time-stepping and preserving orthogonality. *BIT*, 37:24–36, 1997.
- N.J. Higham and R.S. Schreiber. Fast polar decomposition of an arbitrary matrix. *SIAM J. Sci. Stat. Comput.*, 11:648–655, 1990.
- P.L. Houtekamer and H.L. Michell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Mon. Wea. Rev.*, 129:123–136, 2001.

- E. Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2002.
- D.M. Livings, S.L. Dance, and N.K. Nichols. Unbiased ensemble square root filters. *Physica D*, 237:1021–1028, 2008.
- E.N. Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on Predictability*, volume 1, pages 1–18, ECMWF, Reading, Berkshire, UK, 1996.
- E.N. Lorenz and K.E. Emanuel. Optimal sites for supplementary weather observations: Simulations with a small model. *J. Atmos. Sci.*, 55:399–414, 1998.
- D.J. Simon. *Optimal state estimation*. John Wiley & Sons, Inc., New York, 2006.
- M.K. Tippett, J.L. Anderson, G.H. Bishop, T.M. Hamill, and J.S. Whitaker. Ensemble square root filters. *Mon. Wea. Rev.*, 131:1485–1490, 2003.
- Z. Toth and E. Kalnay. Ensemble forecasting at NCEP: The generation of perturbations. *Bull. Amer. Meteor. Soc.*, 74:2317–2330, 1993.
- Z. Toth and E. Kalnay. Ensemble forecasting at NCEP and the breeding method. *Mon. Wea. Rev.*, 125:3297–3319, 1997.
- X. Wang, C.H. Bishop, and S.J. Julier. Which is better, an ensemble of positive-negative pairs or a centered spherical simplex ensemble? *Mon. Wea. Rev.*, 132:1590–1505, 2004.
- M. Wei, Z. Toth, R. Wobus, and Y. Zhu. Initial perturbations based on the ensemble transform (ET) technique in the NCEP global operational forecast system. *Tellus*, 60A:62–79, 2008.
- K. Wright. Differential equations for the analytical singular value decomposition of a matrix. *Numer. Math.*, 63:283–295, 1992.

Appendix

The algorithm of Section 4.2 requires the simultaneous propagation of \mathbf{Y} and \mathbf{M} . Here we describe an alternative factorization algorithm, which is computationally less demanding while maintaining numerical robustness. This algorithm has been used for all numerical experiments conducted in this paper.

We assume that \mathbf{X}_n , its polar factorization $\mathbf{X}_n = \mathbf{Y}_n \mathbf{M}_n$, and an approximative SVD of $\mathbf{M}_n \approx \mathbf{V}_n \mathbf{\Sigma}_n \mathbf{V}_n^T$ are given.

We first propagate the ensemble using the given numerical method, i.e.,

$$\widehat{\mathbf{X}}_{n+1} = \mathbf{X}_n + \Delta t \phi(\mathbf{X}_n; \Delta t). \quad (95)$$

We next update \mathbf{Y}_n using

$$\mathbf{Y}_{n+1} = \widehat{\mathbf{X}}_{n+1} \mathbf{M}_n^{-1} - \mathbf{Y}_n \mathbf{T} \mathbf{S}_n, \quad (96)$$

with $\mathbf{S}_n = \mathbf{S}_n^{\text{sym}} + \mathbf{S}_n^{\text{skew}}$ defined by

$$\mathbf{S}_n^{\text{sym}} = \text{sym} \left\{ \Delta t [\mathbf{Y}_n \mathbf{T}]^T \phi(\mathbf{X}_n; \Delta t) \mathbf{M}_n^{-1} \mathbf{T} \right\} - \frac{1}{2} ([\mathbf{Y}_n \mathbf{T}]^T \mathbf{Y}_n \mathbf{T} - \mathbf{T}) \quad (97)$$

and

$$\mathbf{V}_n^T \mathbf{S}_n^{\text{skew}} \mathbf{V}_n \mathbf{\Sigma}_n + \mathbf{\Sigma}_n \mathbf{V}_n^T \mathbf{S}_n^{\text{skew}} \mathbf{V}_n = \mathbf{V}_n^T \mathbf{M}_n \mathbf{S}_n^{\text{sym}} \mathbf{V}_n - \mathbf{V}_n^T \mathbf{S}_n^{\text{sym}} \mathbf{M}_n \mathbf{V}_n. \quad (98)$$

Here $\text{sym}\{\mathbf{A}\}$ denotes the symmetric part of \mathbf{A} . We next update \mathbf{M}_n using

$$\mathbf{M}_{n+1} = \text{sym} \left\{ [\mathbf{Y}_{n+1} \mathbf{T}]^T \widehat{\mathbf{X}}_{n+1} \mathbf{T} + \mathbf{w} \mathbf{e}^T \right\}. \quad (99)$$

We update the SVD of \mathbf{M}_n by applying two iterations of the (symmetric) SVD post-processing algorithm to \mathbf{M}_{n+1} and finally set

$$\mathbf{X}_{n+1} = \mathbf{Y}_{n+1} \mathbf{M}_{n+1}. \quad (100)$$

Note that the updated ensemble mean satisfies

$$\bar{\mathbf{x}}_{n+1} = \mathbf{X}_{n+1} \mathbf{w} = \widehat{\mathbf{X}}_{n+1} \mathbf{w} = \mathbf{Y}_{n+1} \mathbf{w}. \quad (101)$$

We also have $\mathbf{T} \mathbf{M}_{n+1} = \mathbf{M}_{n+1} \mathbf{T}$, $\mathbf{M}_{n+1} = \mathbf{M}_{n+1}^T$, and the stabilization term in (97) leads to

$$[\mathbf{Y}_{n+1} \mathbf{T}]^T \mathbf{Y}_{n+1} \mathbf{T} \approx \mathbf{T} \quad (102)$$

for Δt sufficiently small.

Under the assumption of $k \gg m$, the computational cost is dominated by 6 matrix-matrix multiplications each of which requires $\mathcal{O}(km^2)$ floating point operations. Such matrix-matrix multiplications can be performed very efficiently on vector or parallel machines (Golub and Loan, 1996).