SIAM J. Applied Dynamical Systems
Vol. 17, No. 4, pp. 2446–2477

Projected Shadowing-Based Data Assimilation*

Bart de Leeuw[†], Svetlana Dubinkina[†], Jason Frank[‡], Andrew Steyer[§], Xuemin Tu[¶], and Erik Van Vleck[¶]

Abstract. In this article we develop algorithms for data assimilation based upon a computational time dependent stable/unstable splitting. Our particular method is based upon shadowing refinement and synchronization techniques and is motivated by work on assimilation in the unstable subspace [Carrassi et al., *Chaos*, 18 (2008), 023112; Trevisan, D'Isidoro, and Talagrand, *Q. J. R. Meteorol. Soc.*, 136 (2010), pp. 487–496; Palatella, Carrassi, and Trevisan, *J. Phys. A*, 46 (2013), 254020] and pseudo-orbit data assimilation [Judd and Smith, *Phys. D*, 151 (2001), pp. 125–141; Judd et al., *J. Atmos. Sci.*, 65 (2008), pp. 1749–1772; Du and Smith, *J. Atmos. Sci.*, 71 (2014), pp. 469–482]. The algorithm utilizes time dependent projections onto the nonstable subspace determined by employing computational techniques for Lyapunov exponents/vectors. The method is extended to parameter estimation without changing the problem dynamics and we address techniques for adapting the method when (as is commonly the case) observations are not available in the full model state space. We use a combination of analysis and numerical experiments (with the Lorenz 63 and Lorenz 96 models) to illustrate the efficacy of the techniques and show that the results compare favorably with other variational techniques.

Key words. data assimilation, tangent space decomposition, shadowing, synchronization

AMS subject classifications. 62M20, 37C50, 34D06, 37M25

DOI. 10.1137/17M1141163

1. Introduction.

Data assimilation methods combine orbits from a dynamical system model with measurement data to obtain an improved estimate for the state of a physical system. In this paper we develop a data assimilation method in the context of a discrete deterministic model

$$x_{n+1} = F_n(x_n), \quad x_n \in \mathcal{R}^d, \quad n = 0, \dots, N-1, \tag{1}$$

where $F_n : \mathcal{R}^d \to \mathcal{R}^d$. In many applications the model is defined by the time-discretization of

[*] Received by the editors July 28, 2017; accepted for publication (in revised form) by E. Sander July 26, 2018; published electronically October 16, 2018.

http://www.siam.org/journals/siads/17-4/M114116.html

Funding: The work of the fourth and sixth authors was supported by NSF grant DMS-1419047. The work of the fifth author was supported by NSF grants DMS-1419069 and DMS-1723066. The work of the first author was partially supported by the research program Mathematics of Planet Earth 2014 EW project 657.014.001, which is financed by the Netherlands Organisation for Scientific Research (NWO).

[†] Centrum Wiskunde & Informatica, P.O. Box 94079, 1090 GB Amsterdam, Netherlands (b.m.de.leeuw@cwi.nl, s.dubinkina@cwi.nl).

[‡] Utrecht University, Mathematical Insitute, P.O. Box 80010, 3508 TA Utrecht, Netherlands (j.e.frank@uu.nl).

[§] University of Kansas, Department of Mathematics, Lawrence, KS 66405. Current address: Sandia National Laboratories, Albuquerque, NM 87185 (asteyer@sandia.gov).

[¶] University of Kansas, Department of Mathematics, Lawrence, KS 66405 (xuemin@ku.edu, erikvv@ku.edu).

an ODE

$$\dot{x} = f(t, x), \quad x(t) \in \mathcal{R}^d, \tag{2}$$

which in turn may be defined as the space-discretization of a PDE (or system of PDEs).

Let the sequence[1] $\{\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_N\}$ be a distinguished orbit of (1), referred to as the *true solution* of the model, and presumed to be unknown. Suppose we are given a sequence of noisy observations y_n related to $\mathcal{X}_n$ via

$$y_n = H\mathcal{X}_n + \xi_n, \qquad y_n \in \mathcal{R}^b, \quad n = 0, \dots, N, \tag{3}$$

where $H : \mathcal{R}^d \to \mathcal{R}^b$, $b \leq d$, is the observation operator, and the noise variables ξ_n are drawn from a normal distribution $\xi_n \sim \mathcal{N}(0, E)$ with zero mean and known observational error covariance matrix E.

Data assimilation is the problem of finding an orbit (or pseudo-orbit; see subsection 2.1) $\boldsymbol{u} = \{u_0, u_1, \dots, u_N\}$, $u_n \in \mathcal{R}^d$, of the model (1), such that the differences $\|y_n - Hu_n\|$, $n = 0, \dots, N$ are small in an appropriately defined sense. This is done with the aim of minimizing the unknown error $\|u_n - \mathcal{X}_n\|$; see, for example, [53, 38]. For instance, well known four-dimensional variational data assimilation (4DVar) aims at finding the optimal initial condition u_0 of (1) to minimize a cost function

$$C_{\text{var}}(u_0; \{y_n\}) = \sum_{n=1}^{N} (y_n - Hu_n)^T E^{-1} (y_n - Hu_n) + \lambda_n (u_n - F_n(u_{n-1})),$$

where the λ_n are Lagrange multipliers to ensure that the sequence $\{u_n\}$ defines an orbit of (1) (see, e.g., [50, 39, 54, 53] and references therein). One drawback of variational data assimilation is that the number of local minima of the cost function increases dramatically with N [5, 42, 47]. This places a practical limit on the length of the assimilation window—the time period over which observations may be assimilated.

We propose a novel data assimilation method that overcomes this drawback: with the proposed method, increasing the length of the assimilation window may in fact lead to a better estimation. Instead of minimizing a cost function, we search for a zero of the *cost operator*

$$G(\boldsymbol{u}) = \begin{pmatrix} G_0(\boldsymbol{u}) \\ G_1(\boldsymbol{u}) \\ \vdots \\ G_{N-1}(\boldsymbol{u}) \end{pmatrix}, \qquad G_n(\boldsymbol{u}) = u_{n+1} - F_n(u_n), \quad n = 0, \dots, N-1, \tag{4}$$

using a contractive iteration started from (a proxy of) complete, noisy observations. This approach is motivated by research on numerical shadowing methods. We stress that, as is the case with 4DVar, our approach attempts to find an exact orbit of (1) consistent with the

[1] In what follows we will adopt the notation $\{\mathcal{X}_n; n = 0, \dots, N\}$ or simply as $\{\mathcal{X}_n\}$ for a discrete orbit. The latter notation is also occasionally employed to denote an infinite sequence.

observations. However, instead of solving directly for the initial condition, we solve for the whole orbit at once.

As stated, our approach assumes the availability of (noisy) observations of the complete state vectors $\mathcal{X}_n$. In other words, we assume that the observation operator H is the identity matrix on $\mathcal{R}^d$. When only partial observations are available, it is necessary to generate a proxy for complete observations. This can be done by some other cheap but inaccurate data assimilation method. For instance, in subsection 6.1 we demonstrate this idea using direct insertion of noisy partial observations into the iteration (1).

Recent efforts [11, 55, 28, 43, 37, 49] to improve speed and reliability of data assimilation specifically address the partitioning of the tangent space into stable, neutral, and unstable subspaces corresponding to Lyapunov vectors associated with negative, zero, and positive Lyapunov exponents, respectively (see subsection 2.2). In particular, Trevisan, d'Isidoro, and Talagrand propose a modification of 4DVar, so-called 4DVar-AUS, in which corrections are applied only in the unstable and neutral subspaces [55, 43]. Research by Pecora and Carroll [44] indicates that when partial observations are sufficient to constrain the unstable subspace, an orbit of the chaotic Lorenz 63 system can be made to converge exponentially in time to a different, driving orbit, hence refining in the stable subspace. Their work has triggered a substantial body of research on the idea of synchronization of chaos. (See review articles by Pecora et al. [46] and Boccaletti et al. [7].)

Motivated by the above, in this paper we propose a new method for data assimilation that utilizes distinct treatments of the dynamics in the stable and nonstable directions. We find a numerical orbit compatible with observations by using Newton's method with updates projected on the nonstable subspace to emphasize the need to stay close to current observations in nonstable directions. In the stable subspace, we ensure that the trajectory is determined by past observations using a forward integration to synchronize the stable components. Although our focus here is on splitting into nonstable and stable components and then applying shadowing refinement and synchronization techniques, respectively, the splitting framework allows for other possibilities. In particular, if the nonstable subspace is relatively low-dimensional this makes applications of techniques such as particle filters appealing. In addition, 4DVar or Kalman filter techniques may be applied to the stable system with the advantage that these techniques are being applied to a system with contractive dynamics. This also allows the split system to be put in a Bayesian data assimilation context.

In the next section we provide relevant background results. In section 3 we describe the sense in which Newton's method is an effective data assimilation algorithm. While effective, the full Newton's iteration can be made more efficient by restricting the updates to just the nonstable tangent directions, as described in section 4. The updates can then be synchronized in the stable directions as shown by the analysis in Appendix A. We provide details of our implementation in section 5. Finally, in section 6 we provide numerical results for the Lorenz 63 model and compare the method to 4DVar for the Lorenz 96 model. We draw conclusions in section 7.

2. Background.

In subsection 2.1 we review concepts from numerical shadowing, in subsection 2.2 we describe the computation of the tangent space splitting used in this paper, and in subsection 2.3 we review synchronization of chaos.

2.1. Numerical shadowing.

An ε-pseudo-orbit is a sequence $\boldsymbol{u} = \{u_0, u_1, \ldots, u_N\}$ satisfying $\|G_n(\boldsymbol{u})\| < \varepsilon$, $n = 0, \ldots, N-1$, where $\|\cdot\|$ is a norm in $\mathcal{R}^d$. For instance, suppose $F \equiv F_n$ is the exact time-τ flow map of an autonomous ODE $\dot{x} = f(x)$. If the components of $\boldsymbol{u}$ are the iterates of a numerical integrator with local truncation error bounded by ε, then these define an ε-pseudo-orbit of F. The shadowing lemma (e.g., Theorem 18.1.2 of [36]) states that in a neighborhood of a hyperbolic set for F, for every $\delta > 0$ there exists $\varepsilon > 0$ such that every ε-pseudo-orbit is δ-shadowed by an orbit of F, i.e., there exists an orbit $\{x_n\}$ satisfying $x_{n+1} = F(x_n)$ such that $\|u_n - x_n\| < \delta$ for all $n = 0, \ldots, N$. We remark that this means that the initial condition $x_0 \neq u_0$ in general. Rigorous bounds on the global error of numerical integrations with respect to a shadowing orbit can be proved. This is done by applying the Newton–Kantorovich theorem to Newton's iteration for $G(\boldsymbol{x}) = 0$ with starting data given by the numerical iterates $\boldsymbol{u}$ on a time interval that is long relative to the characteristic Lyapunov time [6, 31, 30, 14, 15, 56]. Shadowing is an important analysis technique for obtaining global error bounds on the numerical approximation to the solution of differential equations exhibiting chaos.

Numerical shadowing refinement is a residual correction technique that seeks to correct the residual (departure from being a true solution) by solving the error equation without constraining the initial condition. This makes it applicable over long time intervals for problems with positive Lyapunov exponents. When refining using a Gauss–Newton method the linear equations being solved are underdetermined so by using the pseudoinverse the Newton update is the minimum two norm solution correction of the residual. We can view data assimilation in the same vein by interpreting the data as some approximation to the model solution and set it as our goal to find a particular model solution that shadows the data.

With respect to shadowing, the *inverse problem* is to determine an optimal initial condition u_0 for a numerical integration, such that the numerical iterates $\boldsymbol{u}$ δ-shadow a desired orbit of (2). Shadowing refinement (see, e.g., [29]) employs the pseudo-orbit as an initial guess for $G(\boldsymbol{u}) = 0$ and, as opposed to proving the existence of a nearby zero of G, iteratively refines the pseudo-orbit to obtain an improved approximation of a true solution. This is clearly akin to the data assimilation problem.

Shadowing theory has already motivated a practical data assimilation algorithm known as pseudo-orbit data assimilation (PDA); see, for instance, [34, 33, 25] and references therein. For the PDA approach a cost function

$$C_{\mathrm{PDA}} = \frac{1}{2} \sum_{n=0}^{N-1} G_n^T G_n$$

is minimized and the minimization is also initialized from observations. Obviously, the (nonunique) global minimum of C_{PDA} is zero and this value is reached if and only if $G(\boldsymbol{u}) = 0$, that is, if $\boldsymbol{u}$ is any model trajectory. The approach in [25] approximatly minimizes C_{PDA} by taking a fixed number of gradient descent steps starting from observations. This typically yields not an orbit but a (discrete) pseudo-orbit, i.e., the minimizing sequence satisfies $\|u_n - \mathcal{X}_n\| < \varepsilon$ for all $n = 0, \ldots, N$ and some constant ε. The distance between the pseudo-orbit and the manifold of trajectories is then smaller than the distance between observations and the manifold of trajectories. The midpoint of this pseudo-orbit is then used as the initial condition for a trajectory that should be consistent with model and data. PDA has been applied in operational weather models [33] and parameter estimation [51] and as a method for finding reference trajectories for ensemble forecasting [25].

2.2. Tangent subspace decomposition. In this section we review the decomposition of the tangent space into stable, neutral, and strongly unstable subspaces. This decomposition is central to the method described in this paper. Let $\{x_n; n = 0, \ldots, N\}$ denote an orbit of (1). The fundamental matrix equation associated with $\{x_n\}$ is a matrix valued difference equation

$$X_{n+1} = F_n'(x_n)X_n, \quad n = 0, \ldots, N-1, \tag{5}$$

where $X_n \in \mathcal{R}^{d\times d}$. The iterates of (5) become increasingly ill-conditioned as the columns align with the dominant growth direction. To stably estimate X_n, one may introduce a time-discrete QR factorization. Let $X_0 = Q_0R_0$, and write

$$Q_{n+1}R_{n+1} = F_n'(x_n)Q_n \quad \text{for} \quad n = 0, \ldots, N-1, \tag{6}$$

where $F_n'(x_n)Q_n$ is a matrix product of known quantities, and $Q_{n+1}R_{n+1}$ is the QR factorization found using the modified Gram–Schmidt process. Then $X_1 = F_0'(x_0)Q_0R_0 = Q_1R_1R_0$, $X_2 = F_1'(x_1)X_1 = F_1'(x_1)Q_1R_1R_0 = Q_2R_2R_1R_0$, etc. Note that this procedure is well defined for $Q_n \in \mathcal{R}^{d\times p}$ for $p \leq d$ provided $F_n'(x_n)Q_n$ is full rank for all n. The Gram–Schmidt process yields the unique upper triangular $R_n \in \mathcal{R}^{p\times p}$ with positive diagonal elements and, importantly, preserves the ordering of the columns of the Q_n.

The (local) p $(1 \leq p \leq d)$ largest Lyapunov exponents of the orbit $\{x_n\}$ are extracted from the time average of the logarithm of the diagonal of R_n [23]:

$$\lambda_i = \limsup_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} \ln R_n^{(i,i)}, \quad i = 1, \ldots, p.$$

The method of construction ensures $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. Associated with λ_i is a Lyapunov vector $V_n^{(i)}$. The columns of Q_n generally (for most initial conditions) form an orthonormal basis for the Lyapunov vectors at time n (see [16, 17, 18]). The iteration (5) is a generalized power iteration. For each $\ell = 1, \ldots, p$, one finds in the limit $n \to \infty$ that $\text{span}\{V_n^{(1)}, \ldots, V_n^{(\ell)}\} = \text{span}\{Q_n^{(1)}, \ldots, Q_n^{(\ell)}\}$, where $Q_n^{(i)}$ denotes the ith column of Q_n [23, 20, 1, 4]. Positive (negative) λ_i correspond to tangent directions $Q_n^{(i)}$ in which perturbations grow (decay) exponentially. Consequently, if $\lambda_p \geq 0 > \lambda_{p+1}$, then the matrix $Q_n^u = (Q_n^{(1)}, \ldots, Q_n^{(p)})$ provides an orthonormal basis for the nonstable tangent space at X_n. This time dependent tangent space decomposition is motivated by splittings due to integral separation (related to possessing an exponential dichotomy relative to a constant shift in the stability spectrum) and are analogous to having a decomposition via the Oseledets filtration.

We note here that by approximating the nonstable subspace we obtain information (see [28]) that may be used to analyze the error in data assimilation schemes, namely, in terms of the degree to which observations constrain the uncertainty within the nonstable subspace. We remark that the dimension of the unstable subspace may be much less then the total dimension. In Carrassi et al. [12] it is shown that the assimilation in the unstable subspace

(AUS) framework gives good results for a quasi-geostrophic model described in [48]. This model is of dimension 14784, while the unstable subspace has a dimension of 24 [52].

We will use the computed factors $Q_n^u \in \mathcal{R}^{d\times p}$ to construct projection operators onto the nonstable tangent space. The Q_n^u are quantities that can be computed robustly with good forward error analysis properties (under reasonable assumptions closely related to the continuity of Lyapunov exponents with respect to perturbations and the integral separation or integral separation structure; see [1]). In particular, the results in [21, 22, 57, 2] show that the Q_n^u are continuous with respect to errors in $F'(x_n)$ and quantify the error in the Q_n as a function of the separation in growth/decay rates. In our context this ensures the time dependent projection operators $P_n = Q_n^u Q_n^{uT}$ are robust.

2.3. Synchronization.

Pecora and Carroll [44] demonstrated that an orbit of a chaotic dynamical system (the observer) can sometimes be made to synchronize with a second orbit (the driver) of that system, given partial observations of the driver signal. There is a sizeable body of literature on synchronization of chaos, particularly in the field of systems and control theory [44, 32, 28]. The use of synchronization for data assimilation was first applied in [26].

For our purposes, the following coupled driver-response process is appropriate:

$$x_{n+1} = F_n(x_n), \tag{7a}$$

$$z_{n+1} = P_n x_{n+1} + (I - P_n)F_n(z_n), \tag{7b}$$

where the $P_n \in \mathcal{R}^{d\times d}$ are a sequence of appropriately chosen projection matrices. The manifold $\mathcal{S} = \{(x,z) \in \mathcal{R}^d \times \mathcal{R}^d : x = z\}$ is invariant under these dynamics and is called the *synchronization manifold*. When $\mathcal{S}$ attracts a neighborhood of itself, then for z_0 within the basin of attraction, z_n synchronizes with x_n. Defining $w_n = z_n - x_n$, $n = 0, 1, \ldots,$ the transverse dynamics with respect to $\mathcal{S}$ is given by

$$\begin{aligned} w_{n+1} &= P_n x_{n+1} + (I - P_n)F_n(z_n) - F_n(x_n) \\ &= P_n F_n(x_n) + (I - P_n)F_n(z_n) - F_n(x_n) \\ &= (I - P_n)\left[F_n(x_n + w_n) - F_n(x_n)\right] \\ &= (I - P_n)F_n'(x_n)w_n + r_n(w_n), \end{aligned}$$

where $r_n(w)$ is assumed to be of higher order in w. The projectors P_n need to be chosen to ensure asymptotic stability of the origin under the transverse dynamics. From the stability theory of Lyapunov, it is known that if the sequence $\|w_n\|$ converges exponentially to zero for generic initial conditions, then the Lyapunov exponents of the transverse dynamics must necessarily all be negative. Such a necessary condition is argued by Pecora and Carroll in [45]. On the other hand, negativity of the Lyapunov exponents is also sufficient for convergence in a neighborhood of the origin, if F_n' is regular and r_n is at least second order in w. In our application to data assimilation we will choose P_n to project (in an approximate sense) onto the locally nonstable tangent space Q_n^u. Figure 1 illustrates synchronization of the Lorenz 96 model (see [41] and subsection 6.2) using the driver-response system (7a)–(7b) with projection $P_n = Q_n^u Q_n^{uT}$ for increasing dimension of the projection space p. In particular we observe exponential convergence only when p is greater than or equal to the dimension of the nonstable space, with exponential rate of convergence increasing with p.

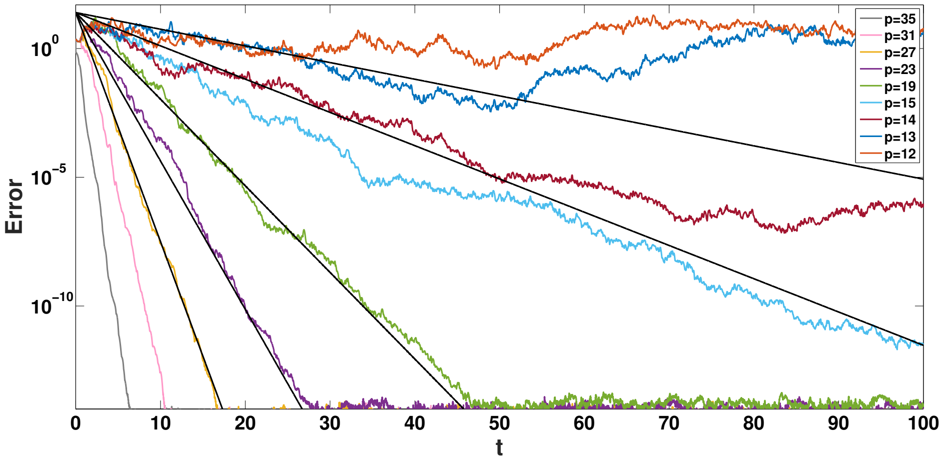

Figure 1. *Synchronization in the L96 model* (21) *with* 13 *positive Lyapunov exponents in* 36 *dimensions. We plot the* ℓ^∞*-norm of the difference between the true solution and the synchronization approximation as a function of time. Forcing is done with projections of the true solution onto the nonstable space and the different graphs are for various values of* p*. In black we plot lines given by* $\mathrm{Error} = 25e^{\lambda_{p+1}\mathrm{Time}}$ *for* $p + 1 = 15, 16, 20, 24, 28$*. It can be observed that after a transient time and for sufficiently large* p *(i.e.,* $p \geq 14$*), convergence to the true solution is exponential.*

3. Data assimilation via Newton's method.

In this section we discuss the use of Newton's method for data assimilation, a context in which it was first applied in [10]. An important property of Newton's method is its local nature: when the initial guess is sufficiently close to a zero, the iterates converge to that zero at a quadratic rate. This statement is made formal in the Kantorovich theorem [35].

Consequently, by analogy to the shadowing approach to global error estimation, we may construct a simple scheme for data assimilation by applying Newton's iterations to solve

$$G(\boldsymbol{u}) = 0,$$

where G is defined in (4) and starting data is provided by the noisy observations $\{y_n : n = 0, \dots, N\}$ with observation operator the identity $\{H\mathcal{X}_n = \mathcal{X}_n : n = 0, \dots, N\}$ (an assumption that can be relaxed; see section 6).

In the kth Newton's iteration we seek an update $\boldsymbol{\delta}^{(k)}$ approximately solving

$$G(\boldsymbol{u}^{(k)} + \boldsymbol{\delta}^{(k)}) = 0. \tag{8}$$

We then update using $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{\delta}^{(k)}$.

The solution to (8) is approximated by iterating

$$G'(\boldsymbol{u}^{(k)})\boldsymbol{\delta}^{(k)} = -G(\boldsymbol{u}^{(k)}), \quad \boldsymbol{u}^{(k+1)} := \boldsymbol{u}^{(k)} + \boldsymbol{\delta}^{(k)} \tag{9}$$

to convergence. We solve each Newton's step using the right pseudoinverse of G', i.e.,

$$G'^{\dagger} = G'^T(G'\,G'^T)^{-1},$$

where the linear system involving the block tridiagonal matrix $G' G'^T$ is solved using a block tridiagonal solver. The function $G(\boldsymbol{u})$ has a zero for every orbit of the model (1). The function $G : \mathcal{R}^{dN} \to \mathcal{R}^{d(N-1)}$ has a $d(N-1) \times dN$ Jacobian with block structure

$$G'(\boldsymbol{u}) = \begin{bmatrix} -F_0'(u_0) & I & & & \\ & -F_1'(u_1) & I & & \\ & & \ddots & \ddots & \\ & & & -F_{N-1}'(u_{N-1}) & I \end{bmatrix}.$$

To distinguish this method from the projected method to be described in section 4, we shall refer to it as the *full Newton's method*.

The fact that Newton's method is a *local* root-finding method proves useful. The Kantorovich theorem can be generalized using the existence of a right inverse as opposed to an inverse. Then the solution $\boldsymbol{u}$ satisfies $G'(\boldsymbol{u})^T G(\boldsymbol{u}) = 0$ [3]. In [34] it is shown that $G'(\boldsymbol{u})^T G(\boldsymbol{u}) = 0$ implies $G(\boldsymbol{u}) = 0$, i.e., $\boldsymbol{u}$ is a trajectory.

Initializing the algorithm with observations, we can expect to find a trajectory close to observations, provided the initial observational error is not too large. The use of the pseudoinverse minimizes the Newton step size in the 2-norm. By taking those shortest possible steps while starting from observations, the iterates should stay close to the observations, while converging to a trajectory. After convergence, it is possible to compute the distance between the assimilation and observations and use that to judge the quality of the assimilations [10, 9, 25].

The convergence of this approach with Newton's method can be demonstrated with numerical examples. Using the Lorenz 63 (L63) model [40] (see also subsection 6.1) a set of true trajectories $\{X_n\}$ were integrated and perturbed. We used 1000 random initial conditions, which were all integrated for a spin-up time of length 5, after which we added noise as specified in (3) with noise covariance $E = I$. Experiments were performed for the L63 model integrated with both a forward Euler scheme and a classic fourth order Runge–Kutta (RK4) scheme. We used time steps of length $\Delta t = 0.005$ for both methods and observed the full state at each time step. For observations sparser in time we refer to subsection 6.2. The perturbed data was used as a starting guess for Newton's method. We used a single observational window of 2000 steps, corresponding to a time interval of length 10. Unless indicated otherwise, all errors are using the ℓ_2-norm.

In Figure 2, we show the median error over time as a dotted line, with the area between the 25th and 75th percentiles shaded. The 2nd and 98th percentiles are shown as solid lines. The results for the forward Euler method are shown in blue and the results for the RK4 method in orange. The mean square error (MSE) is defined as

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^{N} (u_n - \mathcal{X}_n)^T (u_n - \mathcal{X}_n). \tag{10}$$

For the Euler method, the median of the MSE is equal to 0.027. Next we examine errors with respect to the observation operator

$$C(\{x_n\}) = \frac{1}{N} \sum_{n=1}^{N} (y_n - Hx_n)^T (y_n - Hx_n). \tag{11}$$

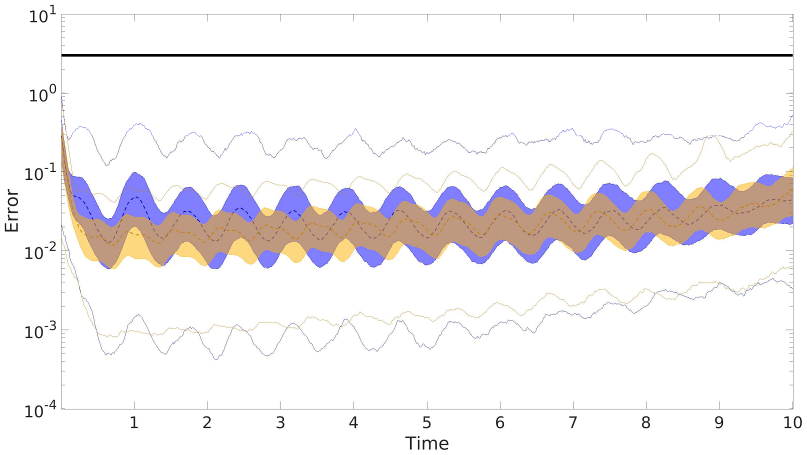

Figure 2. *Application of Newton's method to the L63 model* (19). *The mean observational error is in black and the error over time of the estimation is in blue for forward Euler and orange for $RK4$.*

Table 1
Application of Newton's method to L63. $C(\cdot)$ is defined in (11).

Property	Value
Observation error $C(\{\mathcal{X}_n\})$	2.9994 ± 0.0549
Observational discrepancy of estimated trajectory $C(\boldsymbol{u})$ (Euler)	3.0062 ± 0.0639
Observational discrepancy of estimated trajectory $C(\boldsymbol{u})$ (RK4)	2.9986 ± 0.0550

The MSE in the observations (mean noise variance) is given by $< C(\{\mathcal{X}_n\}) >= 2.9994$, where $< \cdots >$ stands for the average over the experiments. By comparison, the mean observation discrepancy of the Newton's solution is equal to $< C(\boldsymbol{u}) >= 3.0062$ for the Euler discretization experiments and $< C(\boldsymbol{u}) >= 2.9986$ for the RK4 discretization experiments. These results with standard deviations are summarized in Table 1. In 497 out of 1000 Euler experiments and in 860 out of 1000 RK4 experiments, we find that the discrepancy $C(\boldsymbol{u}) < C(\{\mathcal{X}_n\})$. That is, even though the trajectory found by Newton's method is not identical to the true trajectory, it is in fact a model orbit *closer* to the observations. This demonstrates that the method works well, even when observational noise prevents determining a unique viable trajectory.

We have also repeated this numerical experiment for the Lorenz 96 model [41] (see also subsection 6.2). For parameter values $\mathcal{F} = 8$ and $d = 36$, we integrated 1000 initial conditions over a run-up time of 5, after which we assimilated the data taken over a window of length 2.5. We used both forward Euler and RK4 with time step $\tau = 0.005$ and full observations. Results are in Figure 3 and Table 2.

For the Euler discretization of L96 the median of the MSE of the median is 0.0558. In 994 out of 1000 Euler experiments and in 998 out of 1000 RK4 experiments, we find that the discrepancy $C(\boldsymbol{u}) < C(\{\mathcal{X}_n\})$. In the rest of the paper we shall consider results for experiments with Euler discretization. For comparisons, unless stated otherwise, truth and noise realizations are identical between experiments, except in subsection 6.1, where we include results for an ensemble of noise realizations.

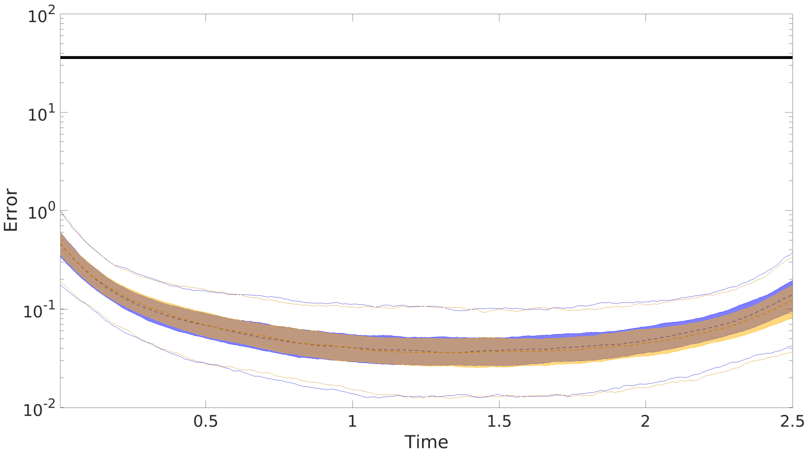

Figure 3. *Application of Newton's method to the L96 model (21). The mean observational error is in black and the error over time of the estimation is in blue for forward Euler and orange for RK4.*

Table 2

Application of Newton's method to L96. $C(\cdot)$ is defined in (11).

Property	Value
Observation error $C(\{\mathcal{X}_n\})$	35.9872 ± 0.3745
Observational discrepancy of estimated trajectory $C(\boldsymbol{u})$ (Euler)	35.9227 ± 0.3751
Observational discrepancy of estimated trajectory $C(\boldsymbol{u})$ (RK4)	35.9229 ± 0.3736

Remark. Our data assimilation method can be applied to parameter estimation as well. A standard approach to dealing with parameter estimation is to treat parameters as dependent variables with trivial dynamics. This approach adds neutral directions to the tangent space, which can hamper convergence of shadowing. Instead, Newton's method can be extended to simultaneously estimate state space variables and parameters. Consider $G(\boldsymbol{u})$ defined in (4) and replace $G_n(\boldsymbol{u})$ with $G_n(\boldsymbol{u};\boldsymbol{\alpha}) = u_{n+1} - F_n(u_n;\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_q)$. Linearization with respect to α_j takes the form

$$\frac{\partial F_n(u_n;\boldsymbol{\alpha})}{\partial \alpha_j} d\alpha_j, \quad n = 0, \dots, N-1.$$

In the presence of uncertain parameters, the linearization of G is modified. In particular, $G'(\boldsymbol{u})$ becomes $G'(\boldsymbol{u};\boldsymbol{\alpha})$. In the case of q parameters we have

$$G'(\boldsymbol{u};\boldsymbol{\alpha}) = [G'_{\boldsymbol{u}} | G'_{\boldsymbol{\alpha}}],$$

where $G_{\boldsymbol{\alpha}}$ is composed of q column vectors. Note that when forming the pseudoinverse of G', $G'^{\dagger} = G'^T(G'G'^T)^{-1}$ we have

$$G'G'^T = G'_{\boldsymbol{u}}G'^T_{\boldsymbol{u}} + G'_{\boldsymbol{\alpha}}G'^T_{\boldsymbol{\alpha}}$$

so that $G'_{\boldsymbol{\alpha}}G'^T_{\boldsymbol{\alpha}}$ is a rank q perturbation of the block tridiagonal matrix $G'_{\boldsymbol{u}}G'^T_{\boldsymbol{u}}$. This allows for the use of Sherman–Morrision–Woodbury formulas and a solver for $G'_{\boldsymbol{u}}G'^T_{\boldsymbol{u}}$ to solve linear systems with matrix $G'G'^T$. In the case of time dependent parameters $\boldsymbol{\alpha}_n$, the diagonal blocks of $G'_{\boldsymbol{u}}G'^T_{\boldsymbol{u}}$ are modified and the overall block tridiagonal structure in maintained in $G'G'^T$. In subsection 6.4 we illustrate this approach with numerical experiments.

4. Tangent space splitting of Newton's method.

In the previous section we demonstrated that Newton's method applied to the residual (4) may converge from noisy observations to a model trajectory. On the other hand the computational and memory costs of the full Newton's method may be high. We will see that when the number of nonnegative Lyapunov exponents is moderate, substantial savings may be realized by computing Newton's updates only in the nonstable directions.

We start by decomposing the relation (8) into the equivalent system

$$\mathcal{P}G\left(\boldsymbol{u}^{(k)} + \widehat{\mathcal{P}}\boldsymbol{\delta}^{(k)} + (I-\widehat{\mathcal{P}})\boldsymbol{\delta}^{(k)}\right) = 0,$$

$$(I-\mathcal{P})G\left(\boldsymbol{u}^{(k)} + \widehat{\mathcal{P}}\boldsymbol{\delta}^{(k)} + (I-\widehat{\mathcal{P}})\boldsymbol{\delta}^{(k)}\right) = 0.$$

Here, $\mathcal{P}$ and $\widehat{\mathcal{P}}$ are block diagonal projection matrices $\mathcal{P} = \text{blockdiag}\,(P_1, \ldots, P_N)$ and $\widehat{\mathcal{P}} = \text{blockdiag}\,(P_0, \ldots, P_N)$, where $P_0, P_1, \ldots, P_N \in \mathcal{R}^{d\times d}$ are projection matrices onto the nonstable subspace at time levels $n = 0, 1, \ldots, N$, respectively.

We propose to modify the Newton's iteration as follows. Instead of computing the update $\boldsymbol{\delta}^{(k)}$ by simultaneously solving the above system, we split the iterate into updates in the range and complement of $\widehat{\mathcal{P}}$. We also allow the projection operators $\mathcal{P}$ and $\widehat{\mathcal{P}}$ to be updated in each iteration. In the kth iteration, we first approximate the update in the range of $\widehat{\mathcal{P}}^{(k)}$, neglecting the term $(I-\widehat{\mathcal{P}}^{(k)})\boldsymbol{\delta}^{(k)}$ in the first equation above and solving

$$\mathcal{P}^{(k)}G\left(\boldsymbol{u}^{(k)} + \widehat{\mathcal{P}}^{(k)}\boldsymbol{\delta}^{(k)}\right) = 0 \tag{12}$$

for ${\boldsymbol{\delta}_{||}}^{(k)} = \widehat{\mathcal{P}}^{(k)}\boldsymbol{\delta}^{(k)}$. Next we approximate the update in the complement of $\widehat{\mathcal{P}}^{(k)}$ by solving

$$(I-\mathcal{P}^{(k)})G(\boldsymbol{u}^{(k)} + \widehat{\mathcal{P}}^{(k)}\boldsymbol{\delta}^{(k)} + (I-\widehat{\mathcal{P}}^{(k)})\boldsymbol{\delta}^{(k)}) = 0 \tag{13}$$

for ${\boldsymbol{\delta}_{\perp}}^{(k)} = (I-\widehat{\mathcal{P}}^{(k)})\boldsymbol{\delta}^{(k)}$. Then the update is computed as $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + {\boldsymbol{\delta}_{||}}^{(k)} + {\boldsymbol{\delta}_{\perp}}^{(k)}$. Expressions (12) and (13) are solved approximately for the components ${\boldsymbol{\delta}_{||}}^{(k)}$ and ${\boldsymbol{\delta}_{\perp}}^{(k)}$ as described below.

4.1. Computation of projection matrices.

The basis Q_n^u, $n = 0, \ldots, N$, for the nonstable tangent space along the true trajectory $\{\mathcal{X}_n\}$ is unknown. Instead, we approximate the Q_n^u along the most recent approximate trajectory $\{u_n^{(k)}\}$. In each iteration we update the projection matrices $P_n^{(k)}$ that project onto the nonstable tangent space. In the kth iteration we choose $P_n^{(k)} = Q_n^{u(k)}(Q_n^{u(k)})^T$, where $Q_n^{u(k)} \in \mathcal{R}^{d\times p}$ is a columnwise orthonormal matrix defined via the iteration (6) linearized along the most recently updated pseudo-orbit $\boldsymbol{u}^{(k)}$. That is, we take $x_n = u_n^{(k)}$, $n = 0, \ldots, N$, in (6). For the first iteration we use the following observations: $u_n^{(0)} = y_n$, $n = 0, \ldots, N$.

The dimension p of the orthonormal basis $Q_n^{u(k)}$ should be equal to or greater than the number of nonnegative Lyapunov exponents. In practice we take p to be a few more than the number of nonnegative Lyapunov exponents to enhance the convergence rate of the synchronization step below [19].

4.2. Newton's step on the unstable space.

Linearization of (12) yields a projected linear system for the update ${\boldsymbol{\delta}_{||}}^{(k)} = \widehat{\mathcal{P}}^{(k)}\boldsymbol{\delta}^{(k)}$:

$$\mathcal{P}^{(k)}G'(\boldsymbol{u}^{(k)})\widehat{\mathcal{P}}^{(k)}\boldsymbol{\delta}^{(k)} = -\mathcal{P}^{(k)}G(\boldsymbol{u}^{(k)}). \tag{14}$$

Suppressing the iteration index k for the moment, define block matrices $\mathcal{Q} = \operatorname{blockdiag}(Q_1^u, \dots, Q_N^u)$ and $\widehat{\mathcal{Q}} = \operatorname{blockdiag}(Q_0^u, \dots, Q_N^u)$ and note the relations $\mathcal{Q}\mathcal{Q}^T = \mathcal{P}$, $\mathcal{Q}^T\mathcal{Q} = I$ with analogous expressions for $\widehat{\mathcal{Q}}$. Let $\boldsymbol{\mu} = \widehat{\mathcal{Q}}^T\boldsymbol{\delta} = \widehat{\mathcal{Q}}^T\widehat{\mathcal{P}}\boldsymbol{\delta}$, $\widetilde{G}' = \mathcal{Q}^T G'(\boldsymbol{u})\widehat{\mathcal{Q}}$ and $\boldsymbol{b} = \mathcal{Q}^T G(\boldsymbol{u})$. Then the linear system for the update $\boldsymbol{\mu}$ may be written as

$$\widetilde{G}'\boldsymbol{\mu} = -\boldsymbol{b}, \tag{15}$$

where the matrix $\widetilde{G}'$ has the block structure

$$\widetilde{G}' = \begin{bmatrix} -R_0^u & I & & & \\ & -R_1^u & I & & \\ & & \ddots & \ddots & \\ & & & -R_{N-1}^u & I \end{bmatrix},$$

and consequently, $\widetilde{G}' \in \mathcal{R}^{Np\times(N+1)p}$. We solve (15) using the right pseudoinverse $\widetilde{G}'^{\dagger} = \widetilde{G}'^T(\widetilde{G}'\widetilde{G}'^T)^{-1}$ and define the intermediate update

$$\bar{\boldsymbol{u}}^{(k)} = \boldsymbol{u}^{(k)} + \widehat{\mathcal{P}}^{(k)}\boldsymbol{\delta}^{(k)} = \boldsymbol{u}^{(k)} + \widehat{\mathcal{Q}}^{(k)}\boldsymbol{\mu}^{(k)}. \tag{16}$$

4.3. Synchronization step in the stable space.

We next turn to the treatment of (13). Inserting the definition (16) into (13) yields the relation

$$(I - \mathcal{P}^{(k)})G(\bar{\boldsymbol{u}}^{(k)} + (I - \widehat{\mathcal{P}}^{(k)})\boldsymbol{\delta}^{(k)}) = 0, \tag{17}$$

whose solution for ${\boldsymbol{\delta}_{\perp}}^{(k)} = (I - \widehat{\mathcal{P}}^{(k)})\boldsymbol{\delta}^{(k)}$ we wish to approximate. Again dropping the iteration index k for the moment, we expand (17) componentwise over the time index n:

$$\begin{aligned} 0 &= \left[(I - \mathcal{P})G\left(\bar{\boldsymbol{u}} + (I - \widehat{\mathcal{P}})\boldsymbol{\delta}\right)\right]_n \\ &= (I - P_{n+1})\left[\bar{u}_{n+1} + (I - P_{n+1})\delta_{n+1} - F_n(\bar{u}_n + (I - P_n)\delta_n)\right], \quad n = 0, \dots, N-1. \end{aligned}$$

The second equation is rewritten in the form

$$(I - P_{n+1})\delta_{n+1} = (I - P_{n+1})\left(F_n(\bar{u}_n + (I - P_n)\delta_n) - \bar{u}_{n+1}\right), \quad n = 0, \dots, N-1.$$

Adding $\bar{u}_{n+1}$ to both sides of this equation we get

$$\bar{u}_{n+1} + (I - P_{n+1})\delta_{n+1} = P_{n+1}\bar{u}_{n+1} + (I - P_{n+1})F_n(\bar{u}_n + (I - P_n)\delta_n), \quad n = 0, \dots, N-1,$$

or, defining $u_n^{(k+1)} = \bar{u}_n^{(k)} + (I - P_n)\delta_n^{(k)}$,

$$u_{n+1}^{(k+1)} = P_{n+1}\bar{u}_{n+1}^{(k)} + (I - P_{n+1})F_n(u_n^{(k+1)}), \quad n = 0, \dots, N-1. \tag{18}$$

The form of this iteration is identical to that of the receiver equation (7b) in the synchronization process. In other words, given the update $\bar{\boldsymbol{u}}^{(k)}$, corrected in the nonstable subspace (16), the correction to the stable subspace can be implemented through a forward synchronization integration (18). In the appendix, we prove that under suitable assumptions, if after k iterations, the error in $\bar{\boldsymbol{u}}^{(k)}$ exists entirely in the stable tangent space in the sense that $P_n\bar{u}_n^{(k)} = \bar{u}_n^{(k)}$ and $\|P_n\mathcal{X}_n - \bar{u}_n^{(k)}\| < \epsilon$, then the forward integration (18) converges exponentially to $\mathcal{X}_n$ as $n \to \infty$.

To summarize, the complete iteration step consists of the following:

1. Compute the approximate basis Q_n^u, $n = 0, \dots, N$, for the tangent bundle along the pseudotrajectory $\{u_n^{(k)}\}$.
2. Solve the linear system (14) for the update $\boldsymbol{\delta}_{\perp}{}^{(k)}$ in the nonstable subspace, and compute the intermediate update $\bar{\boldsymbol{u}}^{(k)}$ from (16).
3. Synchronize in the stable subspace using the forward iteration (18) to obtain $\boldsymbol{u}^{(k+1)}$.

The Newton's step in the unstable subspace is based upon residual ($G_n(\boldsymbol{u}) := u_{n+1} - F_n(u_n)$) correction with both the residual and the correction projected into the unstable subspace. If $P_nG_n(\boldsymbol{u}) = G_n(\boldsymbol{u})$ for all n, i.e, the residual is wholly within the unstable subspace, then the synchronization step in the stable subspace is trivial with $(I - P_n)\delta_n \equiv 0$ for all n. Thus, provided the Newton's iteration converges, all residual correction occurs within the unstable subspace. In the more general case in which the residual is contained, at least for some n, in both the stable and unstable subspaces, then the initialization of the synchronization step makes possible a reduction of the residuals in the stable subspace. This then generates an updated approximate trajectory to linearize about and obtain updated projections. In this case the Newton's step in the unstable subspace may again decrease the residual with respect to these new projections. The process then continues in the updated stable subspace and we continue until the desired tolerance is achieved or the method fails for lack of convergence of the projected Newton's iteration. In general the projected Newton's iteration will converge provided the residuals are small enough as compared to the strength of the hyperbolic structure (exponential dichotomies, etc.) in the projected system.

What we have observed is that better results are obtained by switching after each projected Newton's iterate to the synchronization step as opposed to switching to the synchronization after the projected Newton's has converged to tolerance. We attribute this to the variation in the projections that are produced. We note here that the basic splitting based upon projection into unstable and stable parts allows for different techniques to be employed for each subsystem. It provides a representation for the unstable subspace which we believe will prove useful in assessing the effectiveness and uncertainties in data assimilation techniques. We also emphasize that in contrast with traditional data assimilation techniques but similar to PDA, the only influence of the observations is via the initial guess for the projected Newton's/synchronization scheme. Thus, in a perfect model scenario convergence to a solution depends on the initial guess being within its basin of attraction.

In the next section we demonstrate the algorithm and compare it to 4DVar for a number of test problems.

5. Implementation.

In this section we provide details of the algorithm we implement and discuss some possible variations. The algorithm is "interval sequential" in the sense that the shadowing refinement is applied over an entire subinterval. This has the effect of simultaneously incorporating all observations over this subinterval into a single refinement step. In order to transition between subintervals we impose a continuity constraint in the stable subspace. Also discussed in this section are methods for obtaining an initial approximation of the solution trajectory. This is needed in order to determine the initial projections on each subinterval.

When observations are not available at every time step, we can redefine F to be the map corresponding to the composition of several time steps. (Examples are given in subsections 6.2 and 6.3.) When observations are not of the full model state, we can first apply a preprocessing step to the observations to infer an estimate of the full state at all observation times and then perform the main algorithm with the goal of substantial noise reduction. For the PDA method, where the same issue arises, this completion has been done using a variational analysis [33] or by just inserting climatological means for missing observations [24, 25]. In subsection 6.1, we demonstrate an alternative preprocess motived by synchronization, whereby the observation data is directly inserted as a driving signal. The effectiveness of such an approach relies on the ability of the partial observational data to constrain the unstable tangent space. However, it is one of the main conclusions of this paper that such a requirement on the data must hold anyway, if data assimilation is to be effective.

In our implementation, we decompose the time interval $t \in [0, T]$ of integration into $M+1$ nonoverlapping time windows, and the data assimilation method is applied sequentially on each of these. We identify times τ_m, $m = 0, \dots, M+1$, where $\tau_0 = 0$, $\tau_{M+1} = T$, $\tau_1 \in (0, T)$ is the length of the first time window, and $\tau_m = \tau_1 + (m-1)\Delta\tau$, $\Delta\tau = (T - \tau_1)/M$. The mth time window is the interval $t \in [\tau_{m-1}, \tau_m]$. In each window, an initial condition $(\delta_\perp)_0$ is needed for the synchronization step, and convergence of the stable directions requires this quantity to be small (see the appendix). In particular we implement (18) as

$$(I - P_{n+1})\delta_{n+1} = (I - P_{n+1})[F(\bar{u}_n^{(k)} + (I - P_n)\delta_n) - \bar{u}_{n+1}^{(k)}].$$

The initial condition $(\delta_\perp)_0 = (I - P_0)\delta_0$ on time window m is determined by imposing

$$(I - P_0)\delta_0 = (I - P_0)[v_T - u_0^{(k)}],$$

where v_T is the converged iterate u at the terminal time on the time window $m-1$. Effectively, by imposing continuity in the stable directions during the full assimilation, also across window boundaries, when solving (18) we can define a unique solution. (This analysis point of view is also taken up in [32] and [28].) To obtain a good initial condition for the algorithm we perform smoothing on an initialization window: i.e., we employ the full Newton's algorithm (see section 3) on a short window and start the forced system (18) from there. This also improves the approximation of the unstable directions at the beginning of the window at which the projected method is started.

6. Numerical experiments. In the preceding sections we have outlined a data assimilation method based on a tangent space splitting into stable and nonstable subspaces. As described, the method assumes noisy observations of the full state of the system (i.e., observation operator H the identity map on $\mathcal{R}^d$) at each time step, and no restrictions are placed on the length of the time interval.

In this section we demonstrate the behavior of the method for low-dimensional test problems: the Lorenz models L63 and L96. We study dependence on dimension of the projection operator and window lengths. We compare the method with 4DVar and investigate the approaches for incomplete observations and parameter estimation.

In all experiments, the observations are generated from the truth by adding i.i.d. zero-mean Guassian noise as in (3) with diagonal covariance matrix $E = \nu^2 I$, where ν^2 denotes the variance of the noise process. As a convergence criterion for the projected Newton's method we use that $\frac{\|\boldsymbol{b}\|_2}{\|\boldsymbol{u}\|_2} < 10^{-15}$, where $\boldsymbol{b}$ is the projected residual in (15).

6.1. Dependence on projector in the L63 model.

The well-known Lorenz attractor [40] is a chaotic dynamical system commonly used as a test problem for data assimilation algorithms. The L63 model is

$$\dot{x}_1 = \sigma(x_2 - x_1), \quad \dot{x}_2 = x_1(\rho - x_3) - x_2, \quad \dot{x}_3 = x_1 x_2 - \beta x_3, \tag{19}$$

where $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho = 28$. The Lyapunov exponents of the Lorenz attractor are $\lambda_1 \approx 0.906$, $\lambda_2 = 0$, $\lambda_3 \approx -14.572$.

For the experiments in this section we generate a (single) set of observations computing a trajectory of L63 on $t \in [0, 20]$ with $T = 20$, using time step $\Delta t = 0.005$, and $\nu^2 = 4$. In all experiments in this section we use an assimilation window of length $\Delta\tau = 2.5$.

In section 3 we observed that the full Newton's method successfully assimilates observations into L63. Now, we examine the proposed algorithm with projected Newton's and synchronization. Since the L63 model can be synchronized by coupling of the x_1-variables [44, 32], it is natural instead of computing Lyapunov vectors to try to take $P = P_{x_1}$, hence always projecting on the x_1-coordinate, and to iterate (16) and (18). Errors (10)–(11) are given in Table 3, where it is clear that for our algorithm the choice $P = P_{x_1}$ is insufficient to obtain an orbit that is close to observations. Since the projection operators P generally do not commute with the forward model solution operator F, the projected Newton's method does not yield a projection of the full model solution, which means in particular that there are important differences between our algorithm and synchronization in the sense of [44, 32].

Therefore we consider the projection operator on the subspace spanned by Lyapunov vectors. First, we choose the dimension of the projection operator to be $p = 1$. This means we use Newton's method in the (approximate) unstable direction, but not in the neutral or stable direction, because the L63 model has one positive, one zero, and one negative Lyapunov exponent. This is not sufficient for Newton's method to always converge, since the method works well until $t = 20$ and after that Newton's method diverges. The results up to $t = 20$

Table 3

Application of the algorithm to L63 with $P = P_{x_1}$. Results are unsatisfactory. Please recall C and MSE are defined in (11) and (10), respectively.

Property	Value
Observation error $C(\{X_n\})$	11.9
Distance between estimate and observations $C(\boldsymbol{u})$	367
Error between estimate and the truth MSE	356

Table 4

Application of the algorithm to L63 with P_1. Results for the time up to 20, since after that the algorithm diverges. Please recall C, MSE, and D are defined in (11), (10), and (20), respectively.

Property	Value
Observation error $C(\{X_n\})$	11.9
Distance between estimate and observations $C(\boldsymbol{u})$	12.2
Error between estimate and the truth MSE	0.27
Discontinuity measure at window boundaries D	0.23
Average number of iterations #	8.7

are shown in Table 4, where in addition to errors we also display a measure of discontinuity at window boundaries defined as

$$D = \frac{1}{N}\sum_{n=1}^{N} \max |G_n(\boldsymbol{u})| \tag{20}$$

and the average number of iterations needed for Newton's method to converge is denoted as #. We remark that in principle we could restart the method using full Newton's at $t = 20$ and then continue with $p = 1$.

Next, we choose the dimension of the projection operator to be $p = 2$. This means we apply Newton's method to both the unstable and neutral direction. The results are shown in Table 5 and Figure 4, where it can be seen that the algorithm becomes stable. Thus it is necessary to apply the projected Newton's method to both the unstable and neutral directions in this example. We repeated this numerical experiment for 100 different noise realizations. The case $p = d = 3$ is the full Newton's method, which gives the smallest MSE as was shown in section 3. We remark, however, that in this section initialization (full Newton's) is performed only on the first assimilation window, which reduces computational costs. We conclude that the algorithm is capable of recovering a good approximation of the true trajectory and that this approximation *is* a trajectory of the L63 model.

As mentioned in the beginning of section 6, observations of the full model state are not feasible and thus we need to relax this assumption. Therefore we now assume that the only available observations are of the x_1-coordinate. First, we perform a preprocessing procedure in order to complete the missing observations: we run (7b) with observations of x_1 as driving signal and $H\mathcal{X}_n = P_{x_1}\mathcal{X}_n$, $n = 0,\dots,N$, as coupling. Subsequently, we apply the main algorithm with thus completed observations (which generally contain large errors due to the preprocessing). For the main algorithm we choose $p = 2$. Results are shown in Table 6, where we see that when only one coordinate is observed the error can be reduced and information on other coordinates can be obtained with synchronization as the preprocessing procedure.

6.2. Dependence on window length in the L96 model.

Lorenz [41] proposed the following model as an example of a simple one-dimensional model with features of the atmosphere. The L96 model is

$$\dot{x}_l = -x_{l-2}x_{l-1} + x_{l-1}x_{l+1} - x_l + \mathcal{F}, \qquad (l = 1,\dots,d), \tag{21}$$

where the dimension d and forcing $\mathcal{F}$ are parameters. Cyclic boundary conditions are imposed. We implement the L96 model with the standard parameter choices $d = 36$ and $\mathcal{F} = 8$. The

Table 5

Application of the algorithm to L63 with P_2. Good results are obtained. Please recall C, MSE, and D are defined in (11), (10), and (20), respectively. Numbers shown in the table are averages over 100 noise realizations, together with the corresponding standard deviations.

Property	Value
Observation error $C(\{X_n\})$	12.00 ± 0.17
Distance between estimate and observations $C(\boldsymbol{u})$	12.06 ± 0.18
Error between estimate and the truth MSE	0.09 ± 0.07
Discontinuity measure at window boundaries D	0.29 ± 0.08
Average number of iterations #	6.52 ± 0.15

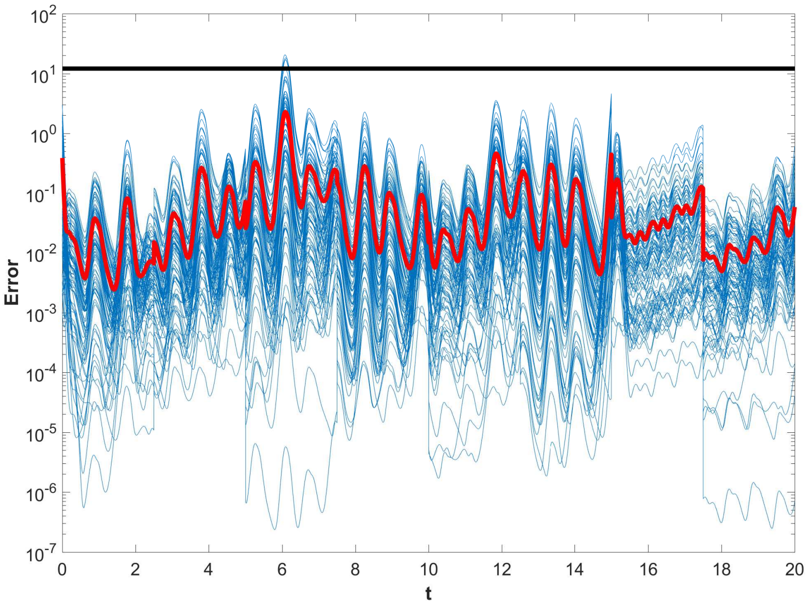

Figure 4. *Application of the algorithm to L63 with P_2. We compare the time-averaged observational error (black) with the error of the estimations (blue) over time. We used 100 observational noise realizations. The average estimation error over time is shown in red.*

Table 6

Application of the algorithm to L63 with P_2 and observations of the x_1-coordinate only. Synchronization is used as a preprocessing step and errors are reduced by the projected Newton's method. Please recall C, MSE, and D are defined in (11), (10), and (20), respectively.

Property	Value
Observation error $C(\{X_n\})$	3.97
Distance between estimate and observations $C(\boldsymbol{u})$	4.32
Error between estimate and the truth MSE	2.49
MSE for the observed coordinate	0.37
Discontinuity measure at window boundaries D	1.16
Average number of iterations #	7.0

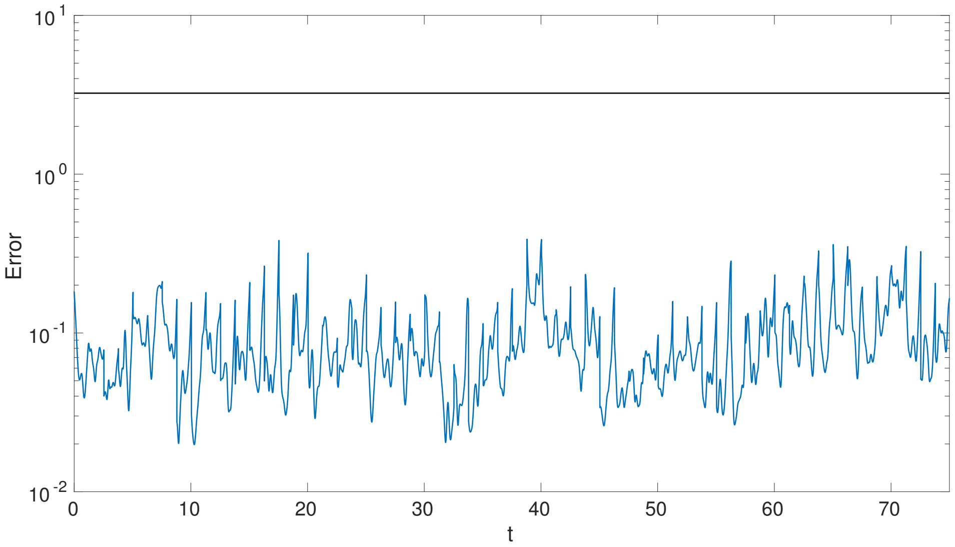

Figure 5. *The best window length choice for the algorithm, applied to L96 with $\sigma = 0.3$ and observations every 10 time steps. We use $p = 15$. The initialization window has length 2.5, and the following windows have length 1.25. In this realization, the $MSE = 0.09$. The mean observational error is in black and the error over time of the estimation is in blue.*

differential equations are discretized with a forward Euler scheme with time step $\tau = 0.005$ and the model initial conditions are chosen at random (standard Gaussian i.i.d.). Observations are obtained by perturbing a reference (true) trajectory with random Gaussian i.i.d. noise with zero mean and covariance with $\sigma = 0.3$. However, the observations are not drawn at every time step as for the L63 model but only every tenth time step, corresponding to observing a full model state every 6 hours. Then the map F_n (1) corresponds to ten forward Euler steps. This map is used to define G and the derivatives of this map are needed for the QR-decompositions and Newton's iteration. For the synchronization we observe that if $G(\boldsymbol{u}) = 0$, then we also have a trajectory under the forward Euler discretization with time step τ. This means that any model integration can just be done with the original discretization, with forcing only applied at points where we have observations.

For the projected Newton's method the dimension of the nonstable subspace p is chosen to be either 15 or 25. We carry out numerical experiments for various choices of window lengths: we use initialization windows with lengths between 0.75 and 15 time units and following windows with lengths between 0.75 and 5 time units. The total time length of assimilation is always 75 and identical observations are used for all choices of p and the respective window length. In Figures 5 and 6 we show the error over time for a particular realization. In Tables 7 and 8 we show mean values and standard deviations for 20 realizations of true trajectory and noise. Initial conditions are generated at random and we use a spin-up time of 5, after which we have observations and assimilation over a total time of 75. It can be seen that the algorithm works well for both long and short windows, although when the windows are too long or too short the results deteriorate. In general, higher p decreases the estimation errors, although for the optimal choice of the window lengths—initialization window of 2.5 and following windows of 1.25—projection on $p = 15$ may result in better estimation; also see Figure 5.

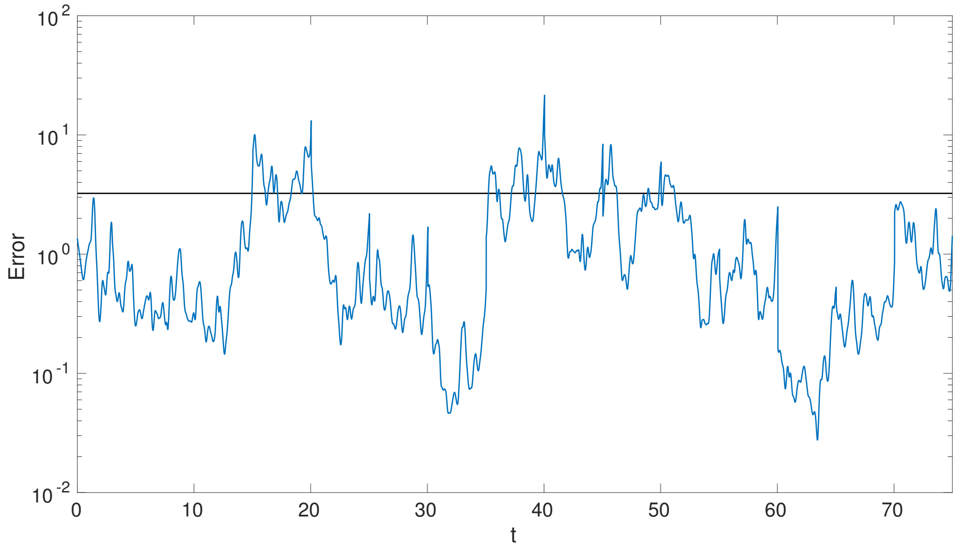

Figure 6. *Here we use $p = 15$ with the longest window lengths: initialization window 15, following windows 5. Trajectories are found, but if error reduction is the main purpose longer windows do not necessarily result in smaller errors. In this realization, the $MSE = 1.47$. The mean observational error is in black and the error over time of the estimation is in blue.*

Table 7

Application of the algorithm to L96 with P_{15}. Please recall C, MSE, and D are defined in (11), (10), and (20), respectively. Averages and standard deviations over 20 trajectory and noise realizations. For window length 5, there were 4 experiments in which the method did not converge. Those results have been left out.

Property	Value			
Window length	5		2.5	
Initial window	15	5	15	5
Observation error $C(\{X_n\})$	3.23 ± 0.01			
Distance between estimate and observations $C(\boldsymbol{u})$	4.64±0.92	4.91±0.71	3.46±0.13	3.33±0.06
Error between estimate and the truth MSE	1.44±0.91	1.71±0.71	0.28±0.13	0.16±0.06
Discontinuity measure at window boundaries D	1.07±0.27	1.05±0.20	0.33±0.05	0.31±0.04
Average number of iterations #	10.6±0.2	10.8±0.3	8.5±0.1	8.5±0.1

Property	Value			
Window length	2.5	1.25		0.75
Initial window	2.5		1.25	0.75
Observation error $C(\{X_n\})$	3.23 ± 0.01			
Distance between estimate and observations $C(\boldsymbol{u})$	3.33±0.04	3.23±0.03	3.25±0.03	3.50±1.09
Error between estimate and the truth MSE	0.16±0.03	0.11±0.03	0.13±0.03	0.46±1.12
Discontinuity measure at window boundaries D	0.31±0.03	0.24±0.03	0.25±0.02	0.31±0.11
Average number of iterations #	8.5±0.1	7.3±0.1	7.4±0.1	7.01±0.04

In Table 9 we investigate in more detail the dependence on the dimension of the projection p. We take total dimension $d = 40$ and illustrate how the distance of the refined orbit from observations undergoes a sharp transition around the number of positive Lyapunov exponents, which is equal to 13. This sharp transition has also been observed for 4DVar-AUS [55, 43].

Table 8

Application of the algorithm to L96 with P_{25}. Please recall C, MSE, and D are defined in (11), (10), and (20), respectively. Averages and standard deviations over 20 trajectory and noise realizations.

Property	Value			
Window length	5		2.5	
Initial window	15	5	15	5
Observation error $C(\{X_n\})$	3.23 ± 0.01			
Distance between estimate and observations $C(\boldsymbol{u})$	4.20±0.33	4.18±0.41	3.37±0.12	3.25±0.03
Error between estimate and the truth MSE	1.00±0.32	0.99±0.41	0.21±0.12	0.10±0.02
Discontinuity measure at window boundaries D	1.01±0.19	0.96±0.17	0.31±0.03	0.30±0.03
Average # of iterations #	9.2±0.2	9.2±0.2	7.9±0.1	7.9±0.1

Property	Value			
Window length	2.5	1.25		0.75
Initial window	2.5		1.25	0.75
Observation error $C(\{X_n\})$	3.23 ± 0.01			
Distance between estimate and observations $C(\boldsymbol{u})$	3.25±0.03	3.15±0.01	3.15±0.01	3.09±0.01
Error between estimate and the truth MSE	0.10±0.02	0.096±0.004	0.098±0.004	0.156±0.004
Discontinuity measure at window boundaries D	0.30±0.03	0.26±0.01	0.26±0.01	0.30±0.01
Average # of iterations #	7.93±0.04	7.01±0.02	7.01±0.02	6.87±0.03

Table 9

Distance between estimate and observations $C(\boldsymbol{u})$ for L96.

p	5	10	13	15	20	30	40
$C(\boldsymbol{u})$	149.2	77.5	73.1	59.2	55.4	53.2	53.2

The values that are reported are based upon uniformly distributed noise in the interval $[-2, 2]$ and upon a total assimilation time of 4, subdivided into 4 windows of length 1. At the first window full Newton's is used (i.e., $p = 40$) and at the subsequent three windows p is as specified in Table 9; $C(\boldsymbol{u})$ is then computed over all 4 windows.

It is possible to carry out the same experiment using more projected windows of length 1 to achieve a total length of 10 or 20. However, for small p, the results get worse if the total length increases. For length 50 divergence of Newton's method is observed. So, the table can be reconstructed qualitatively, but not for unlimited times. At some point new spin-up windows with full Newton's ($p = 40$) are needed. This eventual instability also occurs for L63 with $p = 1$. This problem does not occur when p is chosen large enough. Carrying out the numerical experiment with $p = 20$ results in the average distance to observations remaining compatible with the noise level for time lengths up to 2500 (using windows of length 1).

This illustrates that it is important to define the nonstable space to be large enough, ensuring the $(I - P)$-problem does not contain neutral or unstable directions. If p is chosen too small, the initialization at the spin-up window with full Newton's keeps the error somewhat in check over a few projected windows, but as we progress even further in time projected Newton's on an insufficiently large subspace is unable to keep the error in check. When errors get larger, this will eventually lead to divergence of Newton's method, but already before that the results from the data assimilation get progressively worse. However, if we choose p to be large enough, the method remains stable over long times.

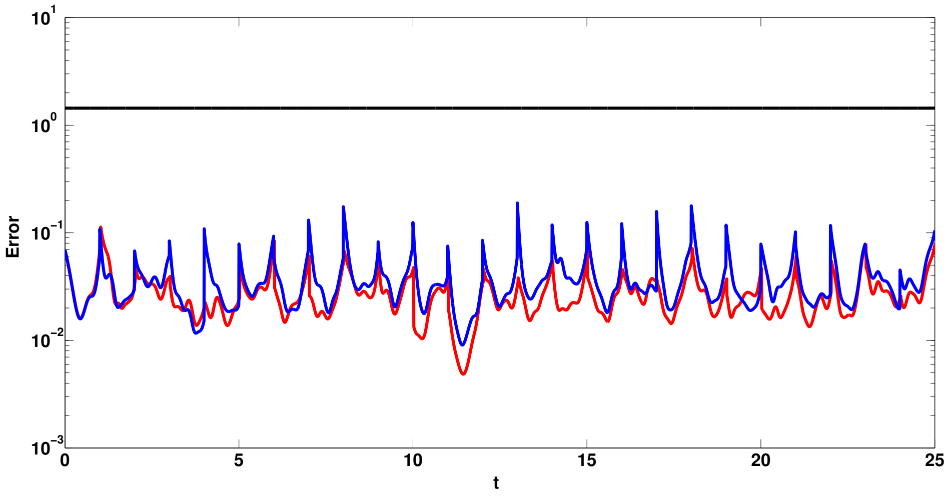

Figure 7. *The error between the results of the approximation methods and the truth over time, measured using the ∞-norm. The error of 4DVar is shown in blue, the error for the shadowing method is shown in red, and the mean observational error is in black.*

Table 10

The results from the projected Newton's algorithm and 4DVar are of comparable quality, but convergence is much more quick for the projected Newton's algorithm.

Property	Value	
Method	Projected Newton's	4DVar
Observation error $C(\{X_n\})$	1.43	
Distance between estimate and observations $C(\boldsymbol{u})$	1.40	1.39
Error between estimate and the truth MSE	0.027	0.037
Discontinuity measure at window boundaries D	0.14	0.17
Average number of iterations #	6.3	418.3

6.3. Comparison to 4DVar.

In the above sections we have argued that our algorithm aims at the same goals as the 4DVar algorithm and that for the L63 and L96 models we are able to reconstruct good trajectories based on observations. We now make a comparison with the standard 4DVar algorithm and demonstrate that our approach is a good alternative.

We perform a test using the L96 model with the same parameters as in the section above. Observations are drawn every fifth time step, which means we observe the full state every 0.025 time units, corresponding to 3 hours. We set $p = 25$. In our tests we use identical data, models, and windows for both methods. We choose 25 windows of length 1, of which the first is used as an initialization window for the shadowing method. On the initialization window 5 iterations are needed for Newton's algorithm to converge. The initialization of 4DVar at the beginning of the first window is done with the first observation, since for neither of the two methods do we have any prior knowledge of the system state. We do not use a background term for 4DVar. The gradient computation in the 4DVar method is done using the adjoint integration and the optimization is performed by a conjugate gradient method. Results are shown in Figure 7 and Table 10.

We can see in Table 10 that the 4DVar method returns a result that is slightly closer to the observations than the projected Newton's method, while the projected Newton's method is slightly closer to the truth. This minor difference might be related to the projection on the unstable space used for the projected Newton's method [55].

It can be seen that 4DVar and the projected Newton's method give comparable results in this test case, but that the projected Newton's method is faster and more suitable if we were to use longer windows. The choice of window length is determined by the requirement that the 4DVar method should still converge. Even though the results in the previous section suggest slightly longer windows would be better, this is not viable for this test. In fact, the number of iterations could be significantly reduced for 4DVar by shortening the window, but this would come at the cost of not taking enough data into account.

If the frequency of observations is significantly lowered, there may be benefits to using 4DVar compared to the projected Newton's method. Since the projected Newton's method makes use of linearizations of the model computed only at observation times, we expect the method to perform worse if observations are far enough apart in time for nonlinearities to be significant. In some cases it may be useful to combine 4DVar and the projected Newton's method. For instance, one could use the solution of the projected Newton's method to initialize the 4DVar algorithm.

We remark that one iteration of the projected Newton's method is more costly than one iteration of the 4DVar algorithm; if most directions are stable, the difference in iteration cost would be less strong. In any case, the higher cost per iteration step of the projected Newton's method is more than compensated for since it requires far fewer iterations. For sufficiently long windows, the cost per iteration of 4DVar is dominated by the need to do one model integration and one adjoint integration, which scales as $\mathcal{O}(Nc)$, where c is defined as the typical cost of taking one time step in the nonlinear model. An implementation of the full Newton's method in which the Jacobians are formed explicitly and are treated as dense matrices yields a cost per iteration of $\mathcal{O}(dNc)$ for the integrations needed and $\mathcal{O}(Nd^3)$ for solving the resulting linear system with a block tridiagonal method. The use of the projected Newton's method reduces this cost to $\mathcal{O}(pNc) + \mathcal{O}(Ndp^2)$. The main factors contributing to the cost per iteration of the projected Newton's method are two model integrations, p tangent linear model integrations and the application of the modified Gram–Schmidt method to a $p \times d$-matrix at each time step.

Reasons for the large difference in number of iterations needed for convergence could be that Newton's method has a quadratic convergence rate, while the optimization algorithm for 4DVar does not. We remark that it is not possible to choose an algorithm with quadratic convergence rate for 4DVar, since we do not have the Hessian of the 4DVar cost function available. A more important reason for the strong difference in number of iterations needed for convergence may be in the fact that 4DVar and projected Newton's really solve very different problems. An explanation for the large difference in needed iterations for this example and the robustness of the projected Newton's method can be found by analyzing what happens when window lengths are increased.

The (projected) Newton's method has to solve larger (but weakly coupled and not that strongly nonlinear) problems if interval length is increased, while for the 4DVar approach the size of the optimization problem stays constant, but the problem becomes more and more

Table 11

Estimation of σ by shadowing-based data assimilation methodology from section 3. The true value is 10.

Property	Initial guess for σ			
Initial σ	5	10	15	20
Observation error $C(\{X_n\})$	2.97			
Distance between estimate and observations $C(\boldsymbol{u})$	2.98	2.96	2.96	2.98
Error between estimate and the truth MSE	0.03	0.02	0.03	0.07
Estimated σ	10.08	10.03	10.05	10.06

Table 12

Estimation of σ by 4DVar parameter estimation. The true value is 10.

Property	Initial guess for σ			
Initial σ	5	10	15	20
Observation error $C(\{X_n\})$	2.97			
Distance between estimate and observations $C(\boldsymbol{u})$	2.98	2.97	3.04	3.06
Error between estimate and the truth MSE	0.12	0.12	0.17	0.18
Estimated σ	9.92	9.95	9.91	9.94

highly nonlinear as the interval length is increased. This problematic behavior of the 4DVar optimization problem is well known in the literature [5, 42, 47]. This can then lead to a large number of iterations needed for convergence, convergence to highly suboptimal local minima, or even to nonconvergence of the 4DVar optimization. This difference between a large but weakly coupled and relatively easy root-finding problem that can be solved with an efficient method compared to a small but highly nontrivial optimization problem for which a slightly slower method has to be employed may give rise to the observed performance difference between the methods, both in terms of iterations needed to converge (and hence time needed to converge) and in the ability to still work for longer windows.

6.4. Parameter estimation. As described in section 3, a shadowing-based data assimilation methodology can be applied to the problem of parameter estimation. The results of σ estimation for the L63 model are shown in Table 11, where different values of initial σ were chosen—5, 10, 15, and 20—with the true σ being 10. Gaussian noise with identity covariance is added to the true solution and data assimilation is performed over one window of length 5. (When data assimilation is performed over multiple windows an estimate from the previous window can be taken as an initial parameter for the next window.) It should be noted that similar results can be obtained for ρ or β estimation of the L63 model and for $\mathcal{F}$ estimation of the L96 model.

In Table 12, we show σ estimations obtained by 4DVar using a window length of 0.25. Instead of using the method of section 3, we can also introduce trivial equations for the parameters to the shadowing-based data assimilation, which introduces extra zero Lyapunov exponents. As can be observed from Table 13 this method fails for σ estimations, though performs sufficiently well for ρ (see Table 14) or β estimations of the L63 model and for $\mathcal{F}$ estimation of the L96 model. Thus, adding trivial equations for the parameters to the shadowing-based data assimilation deteriorates its performance.

Table 13

Estimation of σ by shadowing-based data assimilation methodology with trivial model for the parameters. The true value is 10 *and the estimated σ is the mean estimate. Cases when Newton's method diverges are denoted by "∞" in the corresponding column.*

Property	Initial guess for σ			
Initial σ	5	10	15	20
Observation error $C(\{X_n\})$	2.97			
Distance between estimate and observations $C(\boldsymbol{u})$	360	5.50	∞	∞
Error between estimate and the truth MSE	355	2.77	∞	∞
Estimated σ	8.85	9.83	∞	∞

Table 14

Estimation of ρ by shadowing-based data assimilation methodology with trivial model for the parameters. The true value is 28 *and the estimated ρ is the mean estimate.*

Property	Initial guess for ρ			
Initial ρ	14	28	42	56
Observation error $C(\{X_n\})$	2.97			
Distance between estimate and observations $C(\boldsymbol{u})$	35.7	2.95	5.52	161
Error between estimate and the truth MSE	32.7	0.02	2.74	159
Estimated ρ	26.7	28.0	28.4	37

7. Conclusions.

We have introduced a new class of algorithms for data assimilation based upon shadowing refinement, synchronization, AUS, and PDA techniques. Projections are determined based upon techniques employed in the computation of Lyapunov exponents/vectors, in particular continuous QR techniques. This produces a splitting of the dynamics into nonstable and stable components, which allows for employing different techniques for the different components that are suited to their dynamics. Since the projections are a function of solutions of the state space model, these projection based techniques require at least an approximate solution to determine initial projections. Assessing the uncertainty in obtaining an initial approximate solution and the impact of these uncertainties on the assimilation is a focus of our future work. These techniques are also amenable in a number of ways to a Bayesian framework and since we obtain an approximation of a time dependent orthonormal basis for the nonstable subspace one can assess the observation operator with respect to the unstable subspace. The stable component has contractive dynamics which is useful for error control and further assessing of uncertainties. In future work, we shall analyze the performance of the algorithm when observations are biased, correlated, and/or sparse in time. The algorithm developed here is effective in parameter estimation without introducing a trivial ODE for the parameters as in traditional data assimilation methods. We used a combination of analysis and numerical experiments to show that the algorithm works effectively and we demonstrated that the results compare favorably to those of 4DVar. Rigourous convergence proofs of the method will be addressed in future work. The other avenues for future research include more efficient numerical linear algebra techniques (the shadowing refinement relies on a block tridiagonal linear system solve that we have performed with direct methods) and the use of parallel computing techniques.

Appendix A. Convergence of the synchronization update in the stable subspace.

A.1. Convergence in the linear, nonautonomous case. We study a synchronization process where there is some error made in the nonstable directions. If the model is linear but nonautonomous and at each step sufficiently close to the identity and the largest Lyapunov exponent of the stable subspace is negative, then the total error of the synchronized solution will not be much larger then the error in the nonstable directions. This holds in particular if the largest Lyapunov exponent of the stable subspace is small enough and if convergence to the Lyapunov exponents in the stable space is quick.

Let $\mathcal{X}_n$ be a solution to the nonautonomous linear model $\mathcal{X}_{n+1} = F_n' \mathcal{X}_n$ for $n \in \mathbb{N}$, and let $Q_{n+1} R_{n+1} = F_n' Q_n$. Let $\bar{u}_n \in \mathcal{R}^d$ be a sequence of vectors approximating the truth in the nonstable subspace as follows:

$$P_n \bar{u}_n = \bar{u}_n, \qquad \|P_n \mathcal{X}_n - \bar{u}_n\| < \epsilon,$$

where $P_n := (Q_n^u)(Q_n^u)^T$, the orthogonal projector onto the first p columns of Q_n, i.e., p is the dimension of the nonstable subspace.

Define $\Delta_n := (F_n' - I)$ and $\Delta := \sup_n \|\Delta_n\|_F$. Let w_0 be arbitrary and

$$w_n = (I - P_n) F_{n-1}' \left(\bar{u}_{n-1} + w_{n-1}\right)$$

for $n \geq 1$. Define the error vector v_n as the difference between truth and approximation at time step n:

$$v_n := \mathcal{X}_n - \bar{u}_n - w_n, \tag{22}$$

and denote the projection on the κth column of Q_n by v_n^κ for $\kappa = p+1, \ldots, d$. Let $\zeta := (\frac{\sqrt{2}\Delta}{1-\Delta} + [\frac{\sqrt{2}\Delta}{1-\Delta}]^2)(1+\Delta)\epsilon$ and $\delta^2 = (2\frac{\sqrt{2}\Delta}{1-\Delta}\Delta + [\frac{\sqrt{2}\Delta}{1-\Delta}]^2(1+\Delta))$. Define modified Lyapunov exponents as $\hat{\lambda}_k := \lim_{n\to\infty} \frac{1}{n} \sum_{l=0}^n \log(R_\ell^{(k,k)} + \delta^2)$ for $k = p+1, \ldots, d$.

Assumption A.1. *There exists a positive constant $\hat{\varepsilon} > 0$ and a positive integer $\mathcal{N}$ such that for all integers $\mathcal{N}' \geq \mathcal{N}$ and all $n > 0$,*

$$\left| \frac{1}{\mathcal{N}'} \sum_{\ell=0}^{\mathcal{N}'} \log\left(R_{(n+\ell)}^{(k,k)} + \delta^2\right) - \hat{\lambda}_k \right| < \hat{\varepsilon}, \quad k = p+1, \ldots, d.$$

Theorem A.2. *Assume $\Delta < 1$, $\hat{\lambda}_{p+1} < 0$. Under Assumption A.1, for any $p < \kappa \leq d$, $m_1 \in \mathbb{N}$, and $m_2 := m_1 + \mathcal{N} + 1$,*

$$|v_{m_2}^{(\kappa)}| \leq e^{(\hat{\lambda}_\kappa + \hat{\epsilon}_\kappa)\mathcal{N}} |v_{m_1}^{(\kappa)}| + \sum_{l=0}^{\mathcal{N}} \zeta \prod_{j=l}^{\mathcal{N}-1} \left(R_{j+m_1+2}^{(\kappa,\kappa)} + \delta^2\right)$$
$$+ \left\{ \sum_{\iota=\kappa+1}^{d} \sum_{l=0}^{\mathcal{N}} |R_{l+m_1}^{(\iota,\kappa)} + \delta^2| \prod_{j=l+1}^{\mathcal{N}} \left(R_{j+m_1}^{(\kappa,\kappa)} + \delta^2\right) |v_{l+m_1}^{(\iota)}| \right\}.$$

Corollary A.3. *Assume that when averaging over all $n \in \mathbb{N}$, for all $m \in \mathbb{N}$ and all $\iota_l, \kappa_l \in \{p+1, p+2, \dots, d\}$ with $l \in \{n, n+1, \dots, n+m\}$, the average of the product $\overline{\Pi_{l=n}^{n+m} R_l^{(\iota_l,\kappa_l)}}$ can be expressed as the product of the averages: $\overline{\Pi_{l=n}^{n+m} R_l^{(\iota_l,\kappa_l)}} = \Pi_{l=n}^{n+m} \overline{R_l^{(\iota_l,\kappa_l)}}$. Then*

$$\overline{|v_{m_2}^{(\kappa)}|} < \frac{1}{1 - e^{(\hat{\lambda}_\kappa + \hat{\epsilon}_\kappa)\mathcal{N}}} \left(1 + \frac{1}{|\hat{\lambda}_\kappa|} \left(1 - e^{\hat{\lambda}_\kappa \mathcal{N}} \right) \right) \zeta + \mathcal{O}(\Delta^2),$$

where $\overline{|v_{m_2}^{(\kappa)}|}$ denotes taking the average of $|v_{m_2}^{(\kappa)}|$ over all m_2.

Proof. To prove Theorem A.2 we use, without loss of generality, a coordinate system such that at step n the orthonormal Lyapunov vectors coincide with the standard basis. Then we consider the equation for the projection of the error on the stable space:

$$(I - P_{n+1})v_{n+1} = (I - P_{n+1}) \left[Q_{n+1} R_{n+1} (\mathcal{X}_n - \bar{u}_n - w_n) \right]. \tag{23}$$

We now split the right side of (23) between the range and the kernel of P_n:

$$\begin{aligned}
&(I - P_{n+1}) Q_{n+1} R_{n+1} (\mathcal{X}_n - \bar{u}_n - w_n) \\
&\qquad = (I - P_{n+1}) Q_{n+1} R_{n+1} P_n (\mathcal{X}_n - \bar{u}_n - w_n) \\
&\qquad\quad + (I - P_{n+1}) Q_{n+1} R_{n+1} (I - P_n)(\mathcal{X}_n - \bar{u}_n - w_n) \\
&\qquad = (I - P_{n+1}) Q_{n+1} R_{n+1} (P_n \mathcal{X}_n - \bar{u}_n) + (I - P_{n+1}) Q_{n+1} R_{n+1} \left\{ (I - P_n)\mathcal{X}_n - w_n \right\}.
\end{aligned}$$

We first analyze the contribution from the error term $(I - P_{n+1}) Q_{n+1} R_{n+1} (P_n \mathcal{X}_n - \bar{u}_n)$, which is the contribution of the error parallel to the p leading Lyapunov vectors at time n to the error perpendicular to these vectors at time $n+1$.

By the definitions of Q_n, R_n, and Δ_n, we have that

$$Q_{n+1} R_{n+1} = I + \Delta_n. \tag{24}$$

We recall that $P_n = Q_n^u (Q_n^u)^T$ and that Q_n is the identity matrix in our coordinates. We can approximate $||I - Q_{n+1}||_F \le \frac{\sqrt{2}||\Delta_{n+1}||_F}{1 - ||\Delta_{n+1}||_2}$, by Theorem 3.1 of [13]. It immediately follows that

$$\begin{aligned}
||(I - P_{n+1}) Q_{n+1} R_{n+1} (P_n \mathcal{X}_n - \bar{u}_n)||_2 &< \left(\frac{\sqrt{2}||\Delta_{n+1}||_F}{1 - ||\Delta_{n+1}||_2} + \left[\frac{\sqrt{2}||\Delta_{n+1}||_F}{1 - ||\Delta_{n+1}||_2} \right]^2 \right) (1 + ||\Delta_{n+1}||_2)\epsilon \\
&< \zeta.
\end{aligned}$$

For the convergence in the stable directions we proceed analogously to [56]. We remark that R_n encodes the local approximation to the Lyapunov exponents [27]. We recall that the Gram–Schmidt algorithm ensures that all diagonal elements of R_n are positive for all n. Using the bound on the contribution of the unstable errors to the stable direction, we obtain

$$
\begin{aligned}
||(I-P_{n+1})v_{n+1}||_2 &< ||(I-P_{n+1})F_n'\left\{(I-P_n)v_n\right\}||_2+\zeta \\
&\leq ||(I-P_n)R_{n+1}\left\{(I-P_n)v_n\right\}||_2 \\
&\quad + ||((I-P_{n+1})Q_{n+1}-(I-P_n))R_{n+1}\left\{(I-P_n)v_n\right\}||_2+\zeta \\
&< ||(I-P_n)R_n\left\{(I-P_n)v_n\right\}||_2 \\
&\quad + \left(2\frac{\sqrt{2}||\Delta_{n+1}||_F}{1-||\Delta_{n+1}||_2}\Delta+\left[\frac{\sqrt{2}||\Delta_{n+1}||_F}{1-||\Delta_{n+1}||_2}\right]^2(1+\Delta)\right)||(I-P_n)v_n||_2+\zeta \\
&\leq ||(I-P_n)R_n\left\{(I-P_n)v_n\right\}||_2+\delta^2||(I-P_n)v_n||_2+\zeta.
\end{aligned}
\tag{25}
$$

We now compute a bound in the expected error by induction on the stable subspace dimension $d-p$. If $d-p=1$, then $(I-P_n)v_n=v_n^{(d)}$ and for any m_1, $m_2\in\mathbb{N}$ with $m_2>m_1$,

$$
|v_{m_2}^{(d)}|\leq\prod_{l=m_1}^{m_2-1}\left((R_{dd})_l+\delta^2\right)|v_{m_1}^{(d)}|+\sum_{l=m_1+1}^{m_2}\zeta\prod_{j=l+1}^{m_2}\left((R_{dd})_j+\delta^2\right).
\tag{26}
$$

Using Assumption A.1 and choosing $m_2-m_1=\mathcal{N}+1$, we get

$$
\begin{aligned}
|v_{m_2}^{(d)}| &\leq \prod_{l=0}^{\mathcal{N}}\left((R_{dd})_{l+m_1}+\delta^2\right)|v_{m_1}^{(d)}|+\sum_{l=0}^{\mathcal{N}}\zeta\prod_{j=l}^{\mathcal{N}-1}\left((R_{dd})_{j+m_1+2}+\delta^2\right) \\
&\leq e^{(\hat{\lambda}_d+\hat{\epsilon}_d)\mathcal{N}}|v_{m_1}^{(d)}|+\sum_{l=0}^{\mathcal{N}}\zeta\prod_{j=l}^{\mathcal{N}-1}\left((R_{dd})_{j+m_1+2}+\delta^2\right).
\end{aligned}
$$

Now assume $d-p>1$ and let $p<\kappa\leq d$, and then

$$
\begin{aligned}
|v_{m_2}^{(\kappa)}| &\leq \prod_{l=0}^{\mathcal{N}}\left((R_{\kappa\kappa})_{l+m_1}+\delta^2\right)|v_{m_1}^{(\kappa)}|+\sum_{l=0}^{\mathcal{N}}\zeta\prod_{j=l}^{\mathcal{N}-1}\left((R_{\kappa\kappa})_{j+m_1+2}+\delta^2\right) \\
&\quad+\left\{\sum_{\iota=\kappa+1}^{d}\sum_{l=0}^{\mathcal{N}}|(R_{\iota\kappa})_{l+m_1}+\delta^2|\prod_{j=l+1}^{\mathcal{N}}\left((R_{\kappa\kappa})_{j+m_1}+\delta^2\right)|v_{l+m_1}^{(\iota)}|\right\}.
\end{aligned}
$$

We remark that the first two terms of the above formula are the same as those in (26). This finishes the proof of the theorem.

Corollary A.3 can be proved by taking the average over all m_2. We may now take averages over all $m_2 > \mathcal{N}+1$ and use that $m_1 = m_2 - \mathcal{N} - 1$.

$$\begin{aligned}
\overline{|v^{(d)}_{m_2}|} &\le e^{(\hat{\lambda}_d+\hat{\epsilon}_d)\mathcal{N}}\overline{|v^{(d)}_{m_2-\mathcal{N}-1}|} + \sum_{l=0}^{\mathcal{N}} \zeta \prod_{j=l}^{\mathcal{N}-1} \left(\overline{(R_{dd})_{j+m_2-\mathcal{N}+1}} + \delta^2\right)\\
&\le e^{(\hat{\lambda}_d+\hat{\epsilon})\mathcal{N}}\overline{|v^{(d)}_{m_2}|} + \sum_{l=0}^{\mathcal{N}} \zeta e^{\hat{\lambda}_d l}\\
&< e^{(\hat{\lambda}_d+\hat{\epsilon}_d)\mathcal{N}}\overline{|v^{(d)}_{m_2}|} + \zeta\left(\int_{t=0}^{\mathcal{N}} e^{\hat{\lambda}_d t}\,\mathrm{d}t + 1\right)\\
&\le e^{(\hat{\lambda}_d+\hat{\epsilon}_d)\mathcal{N}}\overline{|v^{(d)}_{m_2}|} + \left(1 + \frac{1}{|\hat{\lambda}_d|}\left(1 - e^{\hat{\lambda}_d\mathcal{N}}\right)\right)\zeta,
\end{aligned}$$

from which it immediately follows that

$$\overline{|v^{(d)}_{m_2}|} < \frac{1}{1 - e^{(\hat{\lambda}_d+\hat{\epsilon}_d)\mathcal{N}}}\left(1 + \frac{1}{|\hat{\lambda}_d|}\left(1 - e^{\hat{\lambda}_d\mathcal{N}}\right)\right)\zeta. \tag{27}$$

For the last term in the expression with $d - p > 1$ this yields

$$\begin{aligned}
&\overline{\sum_{\iota=\kappa+1}^{d}\sum_{l=0}^{\mathcal{N}} |(R_{\iota\kappa})_{l+m_1} + \delta^2| \prod_{j=l+1}^{\mathcal{N}} \left((R_{\kappa\kappa})_{j+m_1} + \delta^2\right)|v^{(\iota)}_{l+m_1}|}\\
&= \sum_{\iota=\kappa+1}^{d}\sum_{l=0}^{\mathcal{N}}\prod_{j=l+1}^{\mathcal{N}} \left(\overline{(R_{\kappa\kappa})_{j+m_1}} + \delta^2\right)\overline{|(R_{\iota\kappa})_{l+m_1} + \delta^2|}\,\overline{|v^{(\iota)}_{l+m_1}|}\\
&< \sum_{\iota=\kappa+1}^{d}\left(1 + \frac{1}{|\hat{\lambda}_\kappa|}\left(1 - e^{\hat{\lambda}_\kappa\mathcal{N}}\right)\right)\left(\Delta + \delta^2\right)\overline{|v^{(\iota)}_{m_2}|},
\end{aligned}$$

from which it immediately follows that

$$\overline{|v^{(\kappa)}_{m_2}|} < \frac{1}{1 - e^{(\hat{\lambda}_\kappa+\hat{\epsilon}_\kappa)\mathcal{N}}}\left(1 + \frac{1}{|\hat{\lambda}_\kappa|}\left(1 - e^{\hat{\lambda}_\kappa\mathcal{N}}\right)\right)\left(\zeta + \sum_{\iota=\kappa+1}^{d}\Delta\overline{|v^{(\iota)}_{m_2}|}\right). \tag{28}$$

Combining (26) and (28) and using that $\zeta = \mathcal{O}(\Delta)$, we conclude that

$$\overline{|v^{(\kappa)}_{m_2}|} < \frac{1}{1 - e^{(\hat{\lambda}_\kappa+\hat{\epsilon}_\kappa)\mathcal{N}}}\left(1 + \frac{1}{|\hat{\lambda}_\kappa|}\left(1 - e^{\hat{\lambda}_\kappa\mathcal{N}}\right)\right)\zeta + \mathcal{O}(\Delta^2). \tag{29}$$

■

A.2. Bound for the nonlinear case. For the nonlinear case we do not have a convergence proof, but we can put a bound on the error. Let the truth $\mathcal{X}_n$ be a solution to the nonlinear model $\mathcal{X}_{n+1} = F_n(\mathcal{X}_n)$, where F_n is a $\mathcal{C}^3$ function. Assume $\{\mathcal{X}_n\}$ lies on an attractor of F and on the attractor F admits an exponential splitting. Let $\epsilon_1 \ge \epsilon_2 > 0$, A_{ϵ_1} be the neighborhood of size ϵ_1 around the attractor of F, $\alpha \geq 0$, $\delta > 0$, and $\tilde{\lambda} > \exp(\lambda_s)$, where $\lambda_s < 0$ is the largest Lyapunov exponent of the stable space. Let Π_n be projectors that project on the nonstable space at $\mathcal{X}_n$, and let $K_2 = \frac{1}{2}\sup_{\chi \in A_{\epsilon_1}} |F_n''(\chi)|$ and $K_0 = \sup_{\chi \in A_{\epsilon_1}} |F_n(\chi)|$. Let $P_n \in \mathcal{R}^d \times \mathcal{R}^d$ be a sequence projectors and let $\bar{u}_n \in \mathcal{R}^d$ be a given sequence of vectors with $P_n \bar{u}_n = \bar{u}_n$ for all n. Let w_0 be some arbitrary vector and $w_n = (I - P_n)F\left(\bar{u}_{n-1} + w_{n-1}\right)$ for $n > 1$. Define the error vector v_n as the difference between truth and approximation at time step n:

$$v_n := \mathcal{X}_n - \bar{u}_n - w_n.$$

Theorem A.4. *Assume* $||v_n|| < \epsilon_1$, $||P_n v_n|| < \epsilon_2$, *and* $||P_{n+1}v_{n+1}|| = ||P_{n+1}\mathcal{X}_{n+1} - \bar{u}_{n+1}|| < \epsilon_2$. *Then there exists some* $\tilde{\alpha} > 0$ *such that if for all* $\tilde{v} \in \mathcal{R}^d$ *it holds that* $P_n\tilde{v} \in K_\alpha^u(\mathcal{X}_n)$, *where* $K_\alpha^u(\mathcal{X}_n)$ *is the nonstable cone* [36, 8] *of size* α *at* $\mathcal{X}_n$, *and* $(I - P_n)\tilde{v} \in K_{\tilde{\alpha}}^s(\mathcal{X}_n)$, *where* $K_{\tilde{\alpha}}^s(\mathcal{X}_n)$ *is the stable cone* [36, 8] *of size* $\tilde{\alpha}$ *at* $\mathcal{X}_n$, *then*

$$||v_{n+1}|| < \epsilon_2 + K_2\epsilon_1^2 + 2\alpha(K_0\epsilon_2 + K_2\epsilon_2^2) + (\tilde{\lambda} + \delta)\epsilon_1. \tag{30}$$

Proof of Theorem A.4. Throughout this proof we use results of [36, 8, 28]. Since the error v_n is small and F_n is $\mathcal{C}^3$, we can approximate the nonlinear flow by a Taylor expansion around the truth

$$||v_{n+1} - F_n'(\mathcal{X}_n)v_n|| = \|F_n(\mathcal{X}_n) - F_n(\mathcal{X}_n - v_n) - F_n'(\mathcal{X}_n)v_n\| \leq K_2||v_n||^2. \tag{31}$$

By splitting $v_{n+1} = P_{n+1}v_{n+1} + (I - P_{n+1})v_{n+1}$, noting that $||I - P_{n+1}|| < 1$, and using (31) we obtain

$$||v_{n+1}|| < \epsilon_2 + K_2\epsilon_1^2 + ||(I - P_{n+1})F_n'(\mathcal{X}_n)v_n||. \tag{32}$$

Due to the exponential splitting of F_n, the nonstable cone becomes more narrow under the tangent dynamics, i.e., vectors in this nonstable cone tend to align more toward the nonstable directions under the dynamics. This means that $F_n'(\mathcal{X}_n)K_\alpha^u(\mathcal{X}_n) \subset \operatorname{int}(K_\alpha^u(\mathcal{X}_{n+1})) \cup \{0\}$. (It follows, for example, from Proposition 5.4.1 of [8] or Lemma 6.2.10 of [36].) Hence we have that $F'(\mathcal{X}_n)P_n v_n \in \operatorname{int}(K_\alpha^u(\mathcal{X}_{n+1})) \cup \{0\}$. Due to the dynamics on the nonstable cone, we expect the length of $P_n v_n$ to grow. A bound for the growth in any step is given by Taylor expansion of $F_n(\mathcal{X}_n - P_n v_n)$ around $F_n(\mathcal{X}_n)$ as $||F'(\mathcal{X}_n)P_n v_n|| \leq K_0||P_n v_n|| + K_2||P_n v_n||^2$, where $K_0 = \sup|F_n(\mathcal{X}_n)|$. However, the only part of $F'(\mathcal{X}_n)P_n v_n$ of interest is the component $(I - P_{n+1})F'(\mathcal{X}_n)P_n v_n$. We have that

$$||(I - P_{n+1})F'(\mathcal{X}_n)P_n v_n|| < 2\alpha||F'(\mathcal{X}_n)P_n v_n|| < 2\alpha(K_0||P_n v_n|| + K_2||P_n v_n||^2). \tag{33}$$

For the part $(I - P_n)v_n$ in the stable cone we can use [8, Proposition 5.4.2] or [36, Lemma 6.2.11], which states that this vector shrinks under time evolution, where the amount depends on the width of our cone. To be precise, for all $\delta > 0$ $\exists\tilde{\alpha} > 0$ such that if $(I - P_n)v_n \in K_{\tilde{\alpha}}^s(\mathcal{X}_n)$, then

$$\|F_n'(\mathcal{X}_n)(I - P_n)v_n\| < (\tilde{\lambda} + \delta)\|(I - P_n)v_n\|. \tag{34}$$

Collecting the estimates (32)–(34), we find that

$$\begin{aligned}||v_{n+1}|| &< \epsilon_2 + K_2\epsilon_1^2 + ||(I - P_{n+1})F_n'(\mathcal{X}_n)(P_n v_n + (I - P_n)v_n)||\\ &\leq \epsilon_2 + K_2\epsilon_1^2 + ||(I - P_{n+1})F_n'(\mathcal{X}_n)P_n v_n|| + ||F_n'(\mathcal{X}_n)(I - P_n)v_n)||\\ &< \epsilon_2 + K_2\epsilon_1^2 + 2\alpha(K_0||P_n v_n|| + K_2||P_n v_n||^2) + (\tilde{\lambda} + \delta)||(I - P_n)v_n||\\ &< \epsilon_2 + K_2\epsilon_1^2 + 2\alpha(K_0\epsilon_2 + K_2\epsilon_2^2) + (\tilde{\lambda} + \delta)\epsilon_1. \qquad \blacksquare\end{aligned}$$

Acknowledgments. The 4DVar experiments were based on a tutorial code of the Reading University Data Assimilation Research Center. We thank Dr. Amos S. Lawless for providing us with this code.

REFERENCES

[1] L. Y. Adrianova, *Introduction to Linear Systems of Differential Equations*, Trans. Math. Monogr. 146, Amer. Math. Soc., Providence, RI, 1995.

[2] M. Badawy and E. Van Vleck, *Perturbation theory for the approximation of stability spectra by QR methods for sequences of linear operators on a Hilbert space*, Linear Algebra Appl., 437 (2012), pp. 37–59, https://doi.org/10.1016/j.laa.2012.01.036.

[3] A. Ben-Israel, *A Newton-Raphson method for the solution of systems of equations*, J. Math. Anal. Appl., 15 (1966), pp. 243 – 252, https://doi.org/10.1016/0022-247X(66)90115-6.

[4] G. Benettin, L. Galgani, A. Giorgilli, and J.-M. Strelcyn, *Lyapunov exponents for smooth dynamical systems and for Hamiltonian systems: A method for computing all of them*, Meccanica, 15 (1980), pp. 9–30.

[5] M. Berliner, *Likelihood and Bayesian prediction for chaotic systems*, J. Amer. Statist. Assoc., 86 (1991), pp. 938–952.

[6] W. Beyn, *On the numerical approximation of phase portraits near stationary-point*, SIAM J. Math. Anal., 24 (1987), pp. 1095–1113, https://doi.org/10.1137/0724072.

[7] S. Boccalettia, J. Kurths, G. Osipov, D. Valladares, and C. Zhou, *The synchronization of chaotic systems*, Phys. Rep., 366 (2002), pp. 1–101.

[8] M. Brin and G. Stuck, *Introduction to Dynamical Systems*, Cambridge University Press, Cambridge, 2002.

[9] J. Bröcker, *On variational data assimilation in continuous time*, Q. J. R. Meteorol. Soc., 136 (2010), pp. 1906–1919.

[10] J. Bröcker and U. Parlitz, *Efficient noncausal noise reduction for deterministic time series*, Chaos, 11 (2001), pp. 319–326.

[11] A. Carrassi, M. Ghil, A. Trevisan, and F. Uboldi, *Data assimilation as a nonlinear dynamical systems problem: Stability and convergence of the prediction-assimilation system*, Chaos, 18 (2008), 023112.

[12] A. Carrassi, A. Trevisan, L. Descamps, O. Talagrand, and F. Uboldi, *Controlling instabilities along a 3DVar analysis cycle by assimilating in the unstable subspace: A comparison with the EnKF*, Nonlinear Process. Geophys., 15 (2008), pp. 503–521.

[13] X.-W. Chang, *On the perturbation of the Q-factor of the QR factorization*, Numer. Linear Algebra Appl., (2012), pp. 607–619, https://doi.org/10.1002/nla.787.

[14] S. Chow, X. Lin, and K. Palmer, *A shadowing lemma with applications to semilinear parabolic equations*, SIAM J. Math. Anal., 20 (1989), pp. 547–557, https://doi.org/10.1137/0520038.

[15] S.-N. Chow and K. J. Palmer, *On the numerical computation of orbits of dynamical systems: The higher dimensional case*, J. Complexity, 8 (1992), pp. 398–423, https://doi.org/10.1016/0885-064X(92)90004-U.

[16] L. Dieci and C. Elia, *The singular value decomposition to approximate spectra of dynamical systems. Theoretical aspects*, J. Differential Equations, 230 (2006), pp. 502–531, https://doi.org/10.1016/j.jde.2006.08.007.

[17] L. Dieci, C. Elia, and E. Van Vleck, *Exponential dichotomy on the real line: SVD and QR methods*, J. Differential Equations, 248 (2010), pp. 287–308, https://doi.org/10.1016/j.jde.2009.07.004.

[18] L. Dieci, C. Elia, and E. Van Vleck, *Detecting exponential dichotomy on the real line: SVD and QR algorithms*, BIT, 51 (2011), pp. 555–579, https://doi.org/10.1007/s10543-010-0306-0.

[19] L. Dieci, R. Russell, and E. Van Vleck, *On the computation of Lyapunov exponents for continuous dynamical systems*, SIAM J. Numer. Anal., 34 (1997), pp. 402–423.

[20] L. Dieci and E. S. Van Vleck, *On the error in computing Lyapunov exponents by QR methods*, Numer. Math., 101 (2005), pp. 619–642.

[21] L. Dieci and E. S. Van Vleck, *Perturbation theory for approximation of Lyapunov exponents by QR methods*, J. Dynam. Differential Equations, 18 (2006), pp. 815–840, https://doi.org/10.1007/s10884-006-9024-3.

[22] L. Dieci and E. S. Van Vleck, *On the error in QR integration*, SIAM J. Numer. Anal., 46 (2008), pp. 1166–1189, https://doi.org/10.1137/06067818X.

[23] L. Dieci and E. S. Van Vleck, *Lyapunov exponents: Computation*, in Encyclopedia of Applied and Computational Mathematics, B. Engquist, ed., Springer, New York, 2015, pp. 834–838.

[24] H. Du, *Combining Statistical Methods with Dynamical Insight to Improve Nonlinear Estimation*, Ph.D. thesis, London School of Economics and Political Science, 2009.

[25] H. Du and L. A. Smith, *Pseudo-orbit data assimilation. Part* I: *The perfect model scenario*, J. Atmos. Sci., 71 (2014), pp. 469–482.

[26] G. S. Duane, J. J. Tribbia, and J. B. Weiss, *Synchronicity in predictive modelling: A new view of data assimilation*, Nonlinear Process. Geophys., 13 (2006), pp. 601–612.

[27] S. V. Ershov and A. B. Potapov, *On the concept of stationary Lyapunov basis*, Phys. D, 118 (1998), pp. 167–198, https://doi.org/10.1016/S0167-2789(98)00013-X.

[28] C. González-Tokman and B. R. Hunt, *Ensemble data assimilation for hyperbolic systems*, Phys. D, 243 (2013), pp. 128–142.

[29] C. Grebogi, S. M. Hammel, J. A. Yorke, and T. Sauer, *Shadowing of physical trajectories in chaotic dynamics: Containment and refinement*, Phys. Rev. Lett., 65 (1990), pp. 1527–1530, https://doi.org/10.1103/PhysRevLett.65.1527.

[30] S. Hammel, J. Yorke, and C. Grebogi, *Numerical orbits of chaotic processes represent true orbits*, Bull. Amer. Math. Soc., 19 (1988), pp. 465–469, https://doi.org/10.1090/S0273-0979-1988-15701-1.

[31] S. M. Hammel, J. A. Yorke, and C. Grebogi, *Do numerical orbits of chaotic dynamical processes represent true orbits?*, J. Complexity, 3 (1987), pp. 136–145, https://doi.org/10.1016/0885-064X(87)90024-0.

[32] K. Hayden, E. Olson, and E. S. Titi, *Discrete data assimilation in the Lorenz and* 2D *Navier–Stokes equations*, Phys. D, 240 (2011), pp. 1416–1425, https://doi.org/10.1016/j.physd.2011.04.021.

[33] K. Judd, C. A. Reynolds, T. E. Rosmond, and L. A. Smith, *The geometry of model error*, J. Atmos. Sci., 65 (2008), pp. 1749–1772.

[34] K. Judd and L. Smith, *Indistinguishable states* I. *Perfect model scenario*, Phys. D, 151 (2001), pp. 125–141.

[35] L. V. Kantorovich and G. P. Akilov, *Functional Analysis in Normed Spaces*, Fizmatgiz, Moscow, 1959.

[36] A. Katok and B. Hasselblatt, *Introduction to the Modern Theory of Dynamical Systems*, Encyclopedia Math. Appl. 54, Cambridge University Press, Cambridge, 1995.

[37] K. Law, D. Sanz-Alonso, A. Shukla, and A. Stuart, *Controlling Unpredictability with Observations in the Partially Observed Lorenz* '96 *Model*, arXiv:1411.3113, 2014.

[38] K. Law, A. Stuart, and K. Zygalakis, *Data Assimilation: A Mathematical Introduction*, Texts Appl. Math. 62, Springer, New York, 2015, https://doi.org/10.1007/978-3-319-20325-6.

[39] J. Lewis and J. Derber, *The use of adjoint equations to solve a variational adjustment problem with advective constraint*, Tellus, 37A (1985), pp. 309–322.

[40] E. N. Lorenz, *Deterministic Nonperiodic Flow*, J. Atmos. Sci., 20 (1963), pp. 130–148.

[41] E. N. Lorenz, *Predictability—A problem partly solved*, in Proceedings of Seminar on Predictability, Vol. 1, T. Palmer and R. Hagedorn, eds., Reading, UK, Cambridge University Press, 1996, pp. 1–18.

[42] R. N. Miller, M. Ghil, and F. Gauthiez, *Advanced data assimilation in strongly nonlinear dynamical systems*, J. Atmos. Sci., 51 (1994), pp. 1037–1056.

[43] L. Palatella, A. Carrassi, and A. Trevisan, *Lyapunov vectors and assimilation in the unstable subspace: Theory and applications*, J. Phys. A, 46 (2013), 254020, http://stacks.iop.org/1751-8121/46/i=25/a=254020.

[44] L. M. Pecora and T. L. Carroll, *Synchronization in chaotic systems*, Phys. Rev. Lett., 64 (1990), pp. 821–824.

[45] L. M. Pecora and T. L. Carroll, *Driving systems with chaotic signals*, Phys. Rev. A, 44 (1991), pp. 2374–2383.

[46] L. M. Pecora, T. L. Carroll, G. A. Johnson, D. J. Mar, and J. F. Heagy, *Fundamentals of synchronization in chaotic systems, concepts, and applications*, Chaos, 7 (1997), pp. 520–543.

[47] C. Pires, R. Vautard, and O. Talagrand, *On extending the limits of variational assimilation in nonlinear chaotic systems*, Tellus, 48A (1996), pp. 96–121.

[48] R. Rotunno and J.-W. Bao, *A case study of cyclogenesis using a model hierarchy*, Monthly Weather Rev., 124 (1996), pp. 1051–1066.

[49] D. Sanz-Alonso and A. M. Stuart, *Long-time asymptotics of the filtering distribution for partially observed chaotic dynamical systems*, SIAM/ASA J. Uncertain. Quantif., 3 (2015), pp. 1200–1220, https://doi.org/10.1137/140997336.

[50] Y. Sasaki, *Some basic formalisms in numerical variational analysis*, Monthly Weather Rev., 98 (1970), pp. 875–883.

[51] L. A. Smith, M. C. Cuéllar, H. Du, and K. Judd, *Exploiting dynamical coherence: A geometric approach to parameter estimation in nonlinear models*, Phys. Lett. A, 374 (2010), pp. 2618–2623.

[52] C. Snyder and T. M. Hamill, *Leading Lyapunov vectors of a turbolent baroclinic jet in a quasi-geostrophic model*, J. Atmos. Sci., 60 (2003), pp. 683–688.

[53] O. Talagrand, *Assimilation of observations, an introduction*, J. Meteorol. Soc. Jpn., 75 (1997), pp. 191–209.

[54] O. Talagrand and P. Courtier, *Variational assimilation of meteorological observations with the adjoint vorticity equation*, Q. J. R. Meteorol. Soc., 113 (1987).

[55] A. Trevisan, M. D'Isidoro, and O. Talagrand, *Four-dimensional variational assimilation in the unstable subspace and the optimal subspace dimension*, Q. J. R. Meteorol. Soc., 136 (2010), pp. 487–496, https://doi.org/10.1002/qj.571.

[56] E. S. Van Vleck, *Numerical Shadowing Using Componentwise Bounds and a Sharper Fixed Point Result*, SIAM J. Sci. Comput., 22 (2001), pp. 787–801.

[57] E. S. Van Vleck, *On the error in the product QR decomposition*, SIAM J. Matrix Anal. Appl., 31 (2009/10), pp. 1775–1791, https://doi.org/10.1137/090761562.