Computing covariant Lyapunov vectors, Oseledets vectors, and dichotomy projectors: A comparative numerical study

Gary Froyland [a,*], Thorsten Hüls [b], Gary P. Morriss [c], Thomas M. Watson [a]

[a] *School of Mathematics and Statistics, University of New South Wales, Sydney NSW 2052, Australia*
[b] *Fakultät für Mathematik, Universität Bielefeld, 33501 Bielefeld, Germany*
[c] *School of Physics, University of New South Wales, Sydney NSW 2052, Australia*

ARTICLE INFO

ABSTRACT

Covariant Lyapunov vectors or Oseledets vectors are increasingly being used for a variety of model analyses in areas such as partial differential equations, nonautonomous differentiable dynamical systems, and random dynamical systems. These vectors identify spatially varying directions of specific asymptotic growth rates and obey equivariance principles. In recent years new computational methods for approximating Oseledets vectors have been developed, motivated by increasing model complexity and greater demands for accuracy. In this numerical study we introduce two new approaches based on singular value decomposition and exponential dichotomies and comparatively review and improve two recent popular approaches of Ginelli et al. (2007) [36] and Wolfe and Samelson (2007) [37]. We compare the performance of the four approaches via three case studies with very different dynamics in terms of symmetry, spectral separation, and dimension. We also investigate which methods perform well with limited data.

1. Introduction

The asymptotic behavior of a linear ODE $\dot{x}(t) = Ax(t)$, $x(t) \in \mathbb{R}^d$ is completely determined by the spectral properties of the $d \times d$ matrix A. Similarly, the long-term behavior of a nonlinear ODE $\dot{x}(t) = f(x(t))$ in a small neighborhood of a fixed point x_0, for which $f(x_0) = 0$, is completely determined by the spectral properties of the linearization of f at x_0. Well-known extensions of these facts can be constructed when x_0 is periodic via Floquet theory. However, for general time-dependent linear ODEs $\dot{x}(t) = A(t)x(t)$, the eigenvalues of $A(t)$ contain no useful information about the asymptotic behavior as the simple example of [1, p. 30] illustrates. On the other hand, if the $A(t)$ are generated by a process with well-defined statistics, there is a good spectral theory for the system $\dot{x}(t) = A(t)x(t)$, and this is the content of the celebrated Oseledets Multiplicative Ergodic Theorem (MET), which we state and explain shortly. The "well-defined statistics" are often generated by some underlying (typically ergodic) dynamical system.

For clarity of exposition, we will discuss discrete-time dynamics; it is trivial to convert a continuous-time system to a discrete-time system by creating e.g. time-1 maps flowing from time t to time $t + 1$. Let X denote our *base space*, the space on which the underlying process that controls the time-dependence of the matrices A occurs. As we will place a probability measure on X, we formally need a σ-algebra $\mathfrak{X}$ of sets that we can measure.[1] We denote the underlying process on X by $T : X \circlearrowleft$ and assume that T is invertible. One formally requires that T is measurable[2] with respect to $\mathfrak{X}$. The "well-defined statistics" are captured by a T-*invariant probability measure* μ on X; that is, $\mu = \mu \circ T^{-1}$, and we say that T preserves μ. Finally, it is common to assume that the underlying process is *ergodic*, which means that any subsets $X' \in \mathfrak{X}$ of X that are invariant ($T^{-1}(X') = X'$, implying that trajectories beginning in X' stay in X' forever in forward time and in backward time) have either μ-measure 0 (they are trivial), or μ-measure 1 (up to sets of μ-measure 0 they are all of X).

---

* Corresponding author. Tel.: +61 2 9385 7050; fax: +61 2 9385 7123.
E-mail addresses: g.froyland@unsw.edu.au (G. Froyland), huels@math.uni-bielefeld.de (T. Hüls), g.morriss@unsw.edu.au (G.P. Morriss), t.watson@unsw.edu.au (T.M. Watson).

[1] For example, if X is a topological space, we can set $\mathfrak{X}$ to be the standard Borel σ-algebra generated by open sets on X.
[2] If X is a topological space, and T is continuous, then T measurable with respect to the standard Borel σ-algebra generated by open sets.

Now we come to the matrices A, which are generated by a measurable matrix-valued function $A : X \to M_d(\mathbb{R})$, where $M_d(\mathbb{R})$ is the space of $d \times d$ real matrices. We choose an initial $x \in X$ and begin iterating T to produce an orbit $x, Tx, T^2x, \ldots$. Concurrently, we multiply $\cdots A(T^2x) \cdot A(Tx) \cdot A(x)$, and we are interested in the asymptotic behavior of this matrix product. In particular, we are interested in (i) the growth rates

$$\lambda(x, v) := \lim_{n \to \infty} \frac{1}{n} \log \|A(T^{n-1}x) \cdots A(Tx) \cdot A(x)v\|$$

as v varies in $\mathbb{R}^d$ and (ii) the subspaces $W(x) \subset \mathbb{R}^d$ on which the various growth rates occur. Throughout, $\|\cdot\|$ denotes the standard Euclidean vector norm or the associated matrix operator norm $\|A\| = \max_{\|v\|=1} \|Av\|$; whether $\|\cdot\|$ is a vector or matrix norm will be clear from the context. Surprisingly, the "well-defined statistics" and ergodicity ensures that these limits exist, and that there are at most d different values $\lambda_1 > \lambda_2 > \cdots > \lambda_\ell \geq -\infty$ that $\lambda(x, v)$ can take, as v varies over $\mathbb{R}^d$ and x varies over μ-almost all of X (note we allow $\lambda_\ell = -\infty$ to include the case of non-invertible A). We can also decompose $\mathbb{R}^d$ pointwise in X as $\mathbb{R}^d = \bigoplus_{i=1}^{\ell} W_i(x)$, where for all $v \in W_i(x) \setminus \{0\}$, one has

$$\lim_{n \to \infty} \frac{1}{n} \log \|A(T^{n-1}x) \cdots A(x)v\| = \lambda_i. \tag{1}$$

The subspaces W_i are *equivariant* (or *covariant*) with respect to A over T; that is, they satisfy

$$W_i(Tx) = A(x)W_i(x) \tag{2}$$

for $1 \leq i < \ell$.

We use the following stronger version of the MET, which guarantees an Oseledets splitting even when the matrices A are non-invertible.

Theorem 1.1 ([2, Theorem 4.1]). *Let T be an invertible ergodic measure-preserving transformation of the probability space $(X, \mathfrak{X}, \mu)$. Let $A : X \to M_d(\mathbb{R})$ be a measurable family of matrices satisfying*

$$\int \log^+ \|A(x)\| \, d\mu(x) < \infty.$$

Then there exist $\lambda_1 > \lambda_2 > \cdots > \lambda_\ell \geq -\infty$ and dimensions $m_1, \ldots, m_\ell$ with $m_1 + \cdots + m_\ell = d$, and a measurable family of subspaces $W_i(x) \subset \mathbb{R}^d$ such that for μ-almost every $x \in X$, the following hold.

1. *$\dim W_i(x) = m_i$,*
2. *$\mathbb{R}^d = \bigoplus_{i=1}^{\ell} W_i(x)$,*
3. *$A(x)W_i(x) \subset W_i(Tx)$ with equality if $\lambda_i > -\infty$,*
4. *For all $v \in W_i(x) \setminus \{0\}$, one has*

$$\lim_{n \to \infty} \frac{1}{n} \log \|A(T^{n-1}x) \cdots A(x)v\| = \lambda_i.$$

The range of applications of the MET to the analysis of dynamical systems is vast. Below, we mention just a few of the settings in which the MET is used.

Example 1.2. 1. *Differentiable dynamics*: One of the first applications of the MET was to differentiable dynamical systems $T : X \circlearrowleft$ on smooth d-dimensional compact manifolds. The matrix function A is the spatial derivative of T, denoted DT. The space $\mathbb{R}^d$ is associated with the tangent space of X and the equivariance condition becomes $W_i(Tx) = DT(x) \cdot W_i(x)$. If T is uniformly hyperbolic, $\bigoplus_{i:\lambda_i > 0} W_i(x) = W^u(x)$, the unstable subspace at $x \in X$ and $\bigoplus_{i:\lambda_i < 0} W_i(x) = W^s(x)$, the stable subspace at x. The spaces $W_i(x)$ provide a refinement of $W^u(x)$ and $W^s(x)$ into subspaces with different growth rates.

2. *Hard disk system*: Consider a fixed number N of hard disks in a region $L_x \times L_y$ moving freely between collisions. In each collision a pair of disks change their velocities [3]. The region may be finite (hard walls) or periodic (toroidal) in either coordinate direction. The quasi-one-dimensional system studied here is a two-dimensional system with L_y less than twice the particle diameter so that the disks remain ordered in the x direction. Here $X = ([0, L_x] \times [0, L_y])^N \times \mathbb{R}^{2N}$ (with the appropriate equivalence classes depending on the choice of hard wall or toroidal boundary conditions) is the collection of $4N$-tuples containing all the coordinates and momenta of the N particles.

The map $T : X \to X, x \mapsto \mathcal{C} \circ \mathcal{F}^{\tau(x)}(x)$ is the composition of a free-flow map $\mathcal{F}^{\tau(x)}$ and a collision map $\mathcal{C}$. The free-flow map moves the disks in straight lines according to their momentum while none of the disks are colliding. The time between collisions is the free-flow time $\tau(x)$ which depends on the initial condition $x \in X$. Collisions occur when the boundary of two disks (or one disk and a wall) touch, and the collision map exchanges velocities along the direction of collision (since all disks are of equal mass). Again, the matrix function A is the spatial derivative of T, so that $A(x) = DT(x) = D\left(\mathcal{C} \circ \mathcal{F}^{\tau(x)}\right)(x)$. Precise details may be found in [4].

3. *PDE*: The Kuramoto–Sivashinski equation is a model for weakly turbulent fluids and flame fronts

$$\eta_t = (\eta^2)_x - \eta_{xx} - \nu\eta_{xxxx},$$

where ν is a damping coefficient. Another familiar example is the complex Ginzburg–Landau equation

$$\eta_t = \eta - (1 + i\beta)|\eta|^2\eta + (1 + i\alpha)\eta_{xx},$$

where $\eta(x, t)$ is complex and α and β are parameters. In both of these cases it is possible to approximate solutions of the partial differential equations using Fourier spectral methods (see [5] for details). For instance, in the case of the 1-dimensional

Kuramoto–Sivashinski PDE we look for solutions of the form

$$\eta(x, t) = \sum_{k=-\infty}^{\infty} a_k(t) e^{ikx/\tilde{L}},$$

where $\tilde{L}$ is a unitless length parameter, then solve the following system of ODEs for the Fourier coefficients $a_k(t)$:

$$\dot{a}_k = (q_k^2 - q_k^4)a_k - i\frac{q_k}{2} \sum_{m=-\infty}^{\infty} a_m a_{k-m},$$

where $q_k = k/\tilde{L}$. Since the a_k decrease rapidly with k, truncating the above system of ODEs is justified.

In the setting of this review we treat X as the space of Fourier coefficients $(a_1, \ldots, a_d)$ of the truncated PDE, and consider the transformation $T : X \to X$ defined by choosing some $\tau > 0$ and letting $T((a_1, \ldots, a_d)) = (a_1(\tau), \ldots, a_d(\tau))$ where the $a_k(t)$ are solutions to the system of ODEs with initial conditions $a_k(0) = a_k$. The matrix function A is again the spatial derivative of T so that

$$A(x) = DT(x) = \begin{pmatrix} \dfrac{\partial a_1(\tau)}{\partial a_1} & \dfrac{\partial a_1(\tau)}{\partial a_2} & \cdots \\ \dfrac{\partial a_2(\tau)}{\partial a_1} & \dfrac{\partial a_2(\tau)}{\partial a_2} & \\ \vdots & & \ddots \end{pmatrix}.$$

4. *Nonautonomous ODEs and transfer operators*: Consider an autonomous ODE $\dot{x}(t) = f(x(t))$ on X (for example, the Lorenz flow on $X = \mathbb{R}^3$), and its flow map $\xi(\tau, x)$ which flows the points x forward τ time units. We think of the coordinates x as a "generalized time" and the ODE $\dot{x}(t) = f(x(t))$ is our base system. We use this base ODE (the driving system) to construct a nonautonomous ODE or skew product ODE as $\dot{y}(t) = F(\xi(x, x), y(t))$. Given an initial time t and a flow time τ, one may construct finite-rank approximations $P_x^{(\tau)}(t)$ of the Perron–Frobenius operator $\mathcal{P}^{(\tau)}(x(t))$ that track the evolution of densities from base "time" $x(t)$ to $x(t + \tau)$; see [6] for details. The matrices $P_x^{(\tau)}(t)$ form a cocycle and Oseledets subspace computations enable the extraction of *coherent sets* in the nonautonomous flow (see [6]). Coherent sets are time-dependent analogues of almost-invariant sets for autonomous systems; see [7–9]. Finite-time constructions for coherent sets are described in [10]. In the setting of this review, $T : X \to X$ is defined as $T(x) = \xi(\tau, x)$, and $A(x) = P_x^{(\tau)}(t)$.

From now on, we denote $A(T^{n-1}x) \cdots A(x)$ as $A(x, n)$. The proof of the classical MET [11] shows that the matrix limit

$$\Psi(x) = \lim_{n \to \infty} \left((A(x, n))^* A(x, n) \right)^{1/2n} \tag{3}$$

exists for μ-almost all $x \in X$. The matrix $\Psi(x)$ is symmetric, depends measurably on x, and its eigenvalues are $e^{\lambda_1(x)} > \cdots > e^{\lambda_\ell(x)}$. The corresponding eigenspaces are denoted $U_1(x), \ldots, U_\ell(x)$ and one has

$$V_i(x) := \bigoplus_{j=i}^{\ell} W_j(x) = \bigoplus_{j=i}^{\ell} U_j(x). \tag{4}$$

Thus $V_i(x)$ captures growth rates from λ_i down to λ_ℓ; that is, for $v \in V_i(x) \setminus V_{i+1}(x)$

$$\lambda_i = \lim_{n \to \infty} \frac{1}{n} \log \|A(x, n)v\|. \tag{5}$$

There are many ways to write the $V_i(x)$ as a direct sum of subspaces $Y_i(x), \ldots, Y_\ell(x)$ such that (5) holds for $v \in Y_i(x)$. Two such ways are shown in (4), but one can inductively choose the $Y_i(x)$ to be *any* space of dimension m_i contained in V_i, with trivial intersection with $V_{i+1}(x)$. Regarding the two possibilities in (5), the "U" decomposition of $V_i(x)$ is orthogonal, while the "W" decomposition (the Oseledets splitting of Theorem 1.1) is equivariant (satisfies property (2)). Thus, $W_i(x)$ is mapped onto $W_i(Tx)$ by $A(x)$, and the subspaces $W_i(x), W_i(Tx), W_i(T^2x), \ldots$, track the evolution of their elements under the linear action of $A(x), A(Tx), A(T^2x), \ldots$. On the other hand, the image of $U_i(x)$ under $A(x)$ is in general *not* mapped onto $U_i(Tx)$ and the same is true of other non-equivariant (or non-covariant) splittings. The splitting given by the $W_i(x)$ is the unique[3] splitting with *both* the correct growth rates *and* the equivariance property.

Until recently, classical results provided existence of the splitting $\mathbb{R}^d = W_1(x) \oplus \cdots \oplus W_\ell(x)$ only in the situation where the matrices $A(x), x \in X$ were invertible (see e.g. Theorem 3.4.11 [13]), and otherwise, one only obtained a filtration (or flag) $\mathbb{R}^d = V_1(x) \supset V_2(x) \supset \cdots \supset V_\ell(x) \supset V_{\ell+1}(x) = \{0\}$ (see e.g. Theorem 3.4.1 [13]). Theorem 1.1 (cf. Theorem 4.1 of [2]) demonstrated that one can remove the invertible matrix hypothesis and still obtain a splitting; we thus in this paper deal directly with the equivariant splitting $\mathbb{R}^d = W_1(x) \oplus \cdots \oplus W_\ell(x)$ as it separates the vectors into individual subspaces directly responsible for each distinct Lyapunov exponent,

---

[3] Uniqueness of the $W_i(x)$ is clear in the case where the matrices $A(x), x \in X$ are invertible as the $W_i(x)$ may be *defined* to be the spaces with the right growth rates in both forward and backward time (in the backward time case, T^{-1} and A^{-1} are used), however the splitting is also unique even when the matrices A are not invertible; see also [12] for an operator setting.

with the additional important dynamical property of equivariance. We call the subspaces of this decomposition "Oseledets subspaces"; these are spanned by "Oseledets vectors" (also known as "covariant Lyapunov vectors"[4]).

Non-covariant subspaces such as the $U_i(x)$ have been studied by many authors. Dieci et al. [14–16] prove convergence rates for SVD and QR methods used to approximate the $U_i(x)$ above under the assumption of integral separation (discussed later in this section), as well as convergence rates for approximations of the Sacker–Sell spectrum. The directional information that arises as a byproduct of common QR-based numerical Lyapunov exponent approximation methods [17–19] has been used extensively to analyze particle systems [4,20–23] and is precisely the above $U_i(x)$ decomposition applied to the time-reversed system (in the case where both the map T and matrices $A(x)$ are invertible) [24].

An alternative approach for finding covariant vectors in non-autonomous systems is based on the so called Sacker–Sell spectrum, cf. [25]. The Sacker–Sell spectrum is defined via exponential dichotomies, cf. [26,27] which we briefly introduce for linear difference equations of the form

$$w_{n+1} = A_n w_n, \quad n \in \mathbb{Z}, \ A_n \in M_d(\mathbb{R}) \text{ invertible}. \tag{6}$$

In the current context we associate the sequence of matrices $\{A_n\}_{n \in \mathbb{Z}}$ with an invertible matrix cocycle over a single orbit, e.g. for some $x \in X$ let $A_n = A(T^n x)$. We restrict the introduction of exponential dichotomies to invertible systems only, and note that a justification of our algorithm for computing dichotomy projectors strongly depends on this assumption. Theory defining exponential dichotomies for non-invertible matrices is contained in e.g. [28]. Numerical experiments indicate that Algorithms 3.1 and 3.2 also apply in the non-invertible case, however, the corresponding analysis is a topic of future research.

We denote by Φ the solution operator of (6), defined as

$$\Phi(n, m) := \begin{cases} A_{n-1} \ldots A_m, & \text{for } n > m, \\ I, & \text{for } n = m, \\ A_n^{-1} \ldots A_{m-1}^{-1}, & \text{for } n < m. \end{cases}$$

Definition 1.3. The linear difference equation (6) has an *exponential dichotomy* with data $(K, \alpha_s, \alpha_u, P_n^s, P_n^u)$ on $J \subset \mathbb{Z}$, if there exist two families of projectors P_n^s and $P_n^u = I - P_n^s$ and constants $K, \alpha_s, \alpha_u > 0$, such that the following statements hold:

$$P_n^s \Phi(n, m) = \Phi(n, m) P_m^s \quad \forall n, m \in J, \tag{7}$$

$$\begin{aligned} \left\| \Phi(n, m) P_m^s \right\| &\leq K e^{-\alpha_s(n-m)} \\ \left\| \Phi(m, n) P_n^u \right\| &\leq K e^{-\alpha_u(n-m)} \end{aligned} \quad \forall n \geq m, \ n, m \in J.$$

Consider the scaled equation

$$w_{n+1} = e^{-\lambda} A_n w_n, \quad n \in \mathbb{Z}. \tag{8}$$

Definition 1.4. The *Sacker–Sell* or *dichotomy spectrum* is defined as

$$\sigma_{\text{ED}} := \{\lambda \in \mathbb{R} : (8) \text{ has no exponential dichotomy on } \mathbb{Z}\}.$$

The complementary set $\mathbb{R} \setminus \sigma_{\text{ED}}$ is called the *resolvent set*.

In the literature, the Sacker–Sell spectrum is often defined as the set $\gamma \in \mathbb{R}^+$, for which the scaled equation

$$w_{n+1} = \frac{1}{\gamma} A_n w_n, \quad n \in \mathbb{Z}, \tag{9}$$

has no exponential dichotomy, see, for example [28]. Note that the Sacker–Sell spectrum with respect to (8) is the logarithm of the Sacker–Sell spectrum with respect to (9).

Under the additional assumptions that A_n and $(A_n)^{-1}$ are uniformly bounded, the Sacker–Sell spectrum consists of at most d disjoint, closed intervals, where d denotes the dimension of the space, cf. [25,29], i.e. there exists an $\ell < d$ such that

$$\sigma_{\text{ED}} = \bigcup_{i=1}^{\ell} [\lambda_i^-, \lambda_i^+], \quad \text{where } \lambda_{i+1}^+ < \lambda_i^- \text{ for } i = 1, \ldots, \ell - 1.$$

It is well known that the Lyapunov spectrum, when it exists, is a subset of the Sacker–Sell spectrum, see [30]. While the Lyapunov spectrum provides information on bounded solutions of (6) from time 0 to time $n \geq 0$, the Sacker–Sell spectrum gives information on bounded solutions from time m to time n. These answers may be different for different initial n because, in contrast to the MET setting, there is no *a priori* stationarity assumption on a base dynamical system generating the matrix cocycle. Note that for $\lambda \in \mathbb{R} \setminus \sigma_{\text{ED}}$ it follows from [27, Lemma 2.7] that the inhomogeneous equation $w_{n+1} = e^{-\lambda} A_n w_n + r_n$ has for every bounded sequence $r_{\mathbb{Z}}$ a unique bounded solution on $\mathbb{Z}$.

Dichotomy projectors of the scaled equation (8) are constant in resolvent intervals $R_i := (\lambda_i^+, \lambda_{i-1}^-), \ i = 1, \ldots, \ell + 1$, where $\lambda_0^- = \infty$ and $\lambda_{\ell+1}^+ = -\infty$, see Fig. 1. We denote these families of projectors by $(P_n^{i,s}, P_n^{i,u})$.

---

[4] In the literature, there is an inconsistent use of the term "Lyapunov vector". In some cases, Lyapunov vectors refer to the subspaces $U_i(x)$, while in others the term refers to the $W_i(x)$. The presence of the adjective "covariant" indicates that one is referring to the equivariant Oseledets vectors $W_i(x)$. We therefore use the expressions "Oseledets vector" or "covariant Lyapunov vector" to refer to the $W_i(x)$ and refrain from using "Lyapunov vector".
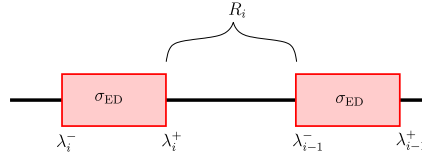
Fig. 1. Spectral setup.

In analogy to the MET we obtain the family of subspaces

$$W_n^i = \mathcal{R}(P_n^{i,s}) \cap \mathcal{R}(P_n^{i+1,u}), \quad n \in \mathbb{Z}, \; i = 1, \dots, \ell$$

that decompose $\mathbb{R}^d$ for each $n \in \mathbb{Z}$

$$\mathbb{R}^d = \bigoplus_{i=1}^{\ell} W_n^i, \tag{10}$$

and using the invariance property (7) it follows for all $i = 1, \dots, \ell$ that

$$A_n W_n^i = W_{n+1}^i, \quad n \in \mathbb{Z}.$$

Furthermore, for each $w \in W_m^i$, there exists a constant $K = K(w) > 0$ such that the following equations hold

$$\|\Phi(n, m)w\| = K e^{\left(\lambda_i^+ + r_i^+(n-m)\right)(n-m)}, \quad \text{for } n \geq m, \text{ where } \limsup_{n \to \infty} r_i^+(n) = 0,$$

$$\|\Phi(n, m)w\| = K e^{\left(\lambda_i^- + r_i^-(n-m)\right)(n-m)}, \quad \text{for } n < m, \text{ where } \limsup_{n \to \infty} r_i^-(n) = 0.$$

Under the assumption of integral separateness one finds d solutions of (6) with pairwise different exponential growth rates [31,32]. Consequently, integrally separated system have d distinct Lyapunov exponents and these exponents are stable w.r.t. additive perturbations of the system, see Theorem 5.4.8 [31]. The latter property is particularly useful for proving error estimates for numerical methods, [14–16]. In contrast to an integral separation, an exponential separation allows a decomposition into $\ell < d$ covariant subspaces $\mathbb{R}^d = \bigoplus_{i=1}^{\ell} W_n^i$, such that each pair of solutions from different subspaces, has a different exponential growth rate; see [33,34]. Exponential separation and integral separation both imply a lower bound on the angle between pairs of subspaces with different growth rates; see [34] and references therein. If a discrete time system has an exponential dichotomy, it is exponentially separated, however, the converse is not true, because, for example, exponential separation may be present, but either exponential expansion or contraction may be absent [34]. In the multiplicative ergodic theorem setting (e.g. Theorem 1.1), one obtains a discrete Lyapunov spectrum with at most d distinct Lyapunov exponents, however without further assumptions, such as exponential separation, in general there is no lower bound on the angle between subspaces in the Oseledets splitting. In the present work, we do not assume simplicity of the covariant splitting (10); i.e. we do not assume that $\dim W_n^i = 1$ for all $i = 1, \dots, d$.

When working with data over a finite time interval, one has access only to a finite sequence $A_0, A_1, \dots, A_{n-1}$. In this case, one either assumes there is an underlying ergodic process generating the sequence $A_0, A_1, \dots, A_{n-1}$ or one considers exponential dichotomies.

An outline of the paper is as follows. In Sections 2 and 3, we introduce two new methods for computing Oseledets vectors. The first method is based on the proof of the generalized MET in [2] and is particularly simple to implement and fast to execute. The second method is an adaptation of an approach to compute dichotomy projectors [35]. In Section 4 we review the approaches by Ginelli et al. [36] and Wolfe and Samelson [37]. In Sections 2–4 we provide MATLAB code snippets to implement the algorithms presented. Section 5 contains numerical comparisons of the performance of the four methods on three dynamical systems. The first case study is a dynamical systems formed via composition of a sequence of 8×8 matrices constructed so that all Oseledets vectors are known at time 0; we thus compare the accuracy of the methods *exactly* in this case study. The second case study is an eight-dimensional system generated by two hard disks in a quasi-one-dimensional box. The third case study is a nonlinear model of time-dependent fluid flow in a cylinder; the matrices are generated by finite-rank approximations of the corresponding time-dependent transfer operators. The three case studies have been chosen to represent a cross-section of a variety of features of systems that either help or hinder the computation of Oseledets vectors, and we draw out the advantages and disadvantages of each of the four methods considered.

2. An SVD-based approach

The approach outlined in this section is simple to execute and exhibits quick convergence. However, as the length of the sample orbit becomes too large this approach fails.

In [2, proof of Theorem 4.1] it is proven that the limit

$$\lim_{N \to \infty} A(T^{-N}x, N) U_i(T^{-N}x)$$

exists and is equal to the ith Oseledets subspace $W_i(x)$. That is, if one computes U_i in the far past and pushes forward to the present, the result is a subspace close to $W_i(x)$. Thus, the strategy in [2] is to first estimate U_i in the past and push forward.

The numerical method of approximating $W_j(x)$, $x \in X$, is implemented in the following steps:

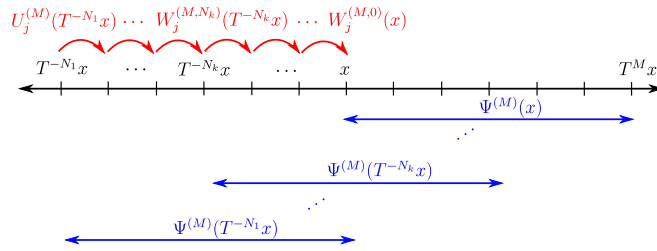**Algorithm 2.1** (*To Estimate $W_j(x)$*).

Fig. 2. Schematic of the re-orthogonalization described in Section 2.1. The black line represents the orbit centered at $x \in X$ and the points $T^{-N_k}x$ are those points at which we ensure orthogonality with the subspaces $V_j(T^{-N_k}x)^\perp$. To do this we use the (blue) approximations $\Psi^{(M)}(T^{-N_k}x)$ to approximate $V_j(T^{-N_k}x)^\perp$ and perform the (red) push-forward and orthogonalization steps starting with $U_j^{(M)}(T^{-N_1}x)$ and ending with $W_j^{(M,0)}(x)$ (see Algorithm 2.2). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

1. Choose $M, N > 0$ and form the matrix

$$\Psi^{(M)}(T^{-N}x) = \left(A(T^{-N}x, M)^* A(T^{-N}x, M)\right)^{1/2M} \tag{11}$$

   as an approximation of (3) at $T^{-N}x \in X$.
2. Compute $U_j^{(M)}(T^{-N}x)$, the jth orthonormal eigenspace of $\Psi^{(M)}(T^{-N}x)$ as an approximation of $U_j(T^{-N}x)$.
3. Define $W_j^{(M,N)}(x) = A(T^{-N}x, N)U_j^{(M)}(T^{-N}x)$, approximating the Oseledets subspace $W_j(x)$.

Listing 1 shows part of a MATLAB implementation of Algorithm 2.1. The array $\mathtt{A} = \left[A(T^{-N}x) \mid A(T^{-N+1}x) \mid \cdots \mid A(T^{M-1}x)\right]$ contains the $d \times d$ matrices which generate the cocycle $A : X \times \mathbb{Z}^+ \to M_d(\mathbb{R})$, and the matrix $\mathtt{Psi}$ is formed by multiplying the matrices contained in $\mathtt{A}$. Step 1 of Algorithm 2.1 is performed prior to the code in Listing 1, Step 2 is performed in lines 1–3 and lines 4–7 perform Step 3. The function returns $\mathtt{Wj}$ as its estimate to $W_j(x)$.

Listing 1: Sample MATLAB code of Algorithm 2.1 to approximate $W_j(x)$.

```
1  [~,s,v] = SVD(Psi);
2  [~,p] = SORT(DIAG(s),'descend');
3  Wj = v(:,p(j))/NORM(v(:,p(j)));
4  FOR h = 1:N
5      Wj = A(:,(h−1)*dim+1:h*dim)*Wj;
6      Wj = Wj/NORM(Wj);
7  END
```

The values of M and N can be chosen with relative freedom and in our examples that follow we have chosen $M = 2N$ to compute over a time window centered on x, from $T^{-N}x$ to $T^N x$. Unfortunately, we cannot choose M and N arbitrarily large and expect accurate results. If $A(T^{-N}x, M)$ is constructed via the product

$$A(T^{-N}x, M) = A(T^{M-N-1}x) \cdots A(T^{-N+1}x)A(T^{-N}x)$$

then with larger M the numerical inaccuracies of matrix multiplication compound and this product becomes more singular and thus a poorer approximation of $A(T^{-N}x, M)$. Because of this, $\Psi^{(M)}(T^{-N}x)$ cannot be expected to accurately approximate $\Psi(T^{-N}x)$ for large M. However, even if we suppose $\Psi^{(M)}(T^{-N}x)$ accurately approximates $\Psi(T^{-N}x)$, the small, but non-zero, difference in $U_j^{(M)}(T^{-N}x)$ and $U_j(T^{-N}x)$ grows roughly as $O\left(e^{N(\lambda_1 - \lambda_j)}\right)$ during the push-forward in Step 3 above. For these reasons M and N must be chosen carefully.

2.1. Improving the basic SVD-based approach

We present a simple improvement that can overcome one of the sources of numerical instability, namely the push-forward process in Step 3 above (see Fig. 2).

Recall that the subspace $V_j(x) = U_j(x) \oplus \cdots \oplus U_\ell(x) = \left(U_1(x) \oplus \cdots \oplus U_{j-1}(x)\right)^\perp$ is A-invariant and that for $v \in V_j(x) \setminus V_{j+1}(x)$ (with $V_{\ell+1}(x) = \{0\}$) we have

$$\lambda_j(x) = \lim_{n \to \infty} \frac{1}{n} \log \|A(x, n)v\| .$$

The subspace $V_j(x)$ contains $W_j(x), W_{j+1}(x), \ldots, W_\ell(x)$, and so the Oseledets subspace $W_j(x)$ is necessarily perpendicular to all $U_1(x), \ldots, U_{j-1}(x)$. To solve the numerical instability of Step 3 we enforce this condition periodically.

The amended algorithm is implemented as follows:

Algorithm 2.2 (*To Estimate $W_j(x)$*).

1. Choose $M, N_1 > N_2 > \cdots > N_n = 0$ and form the matrices

$$\Psi^{(M)}(T^{-N_k}x) = \left(A(T^{-N_k}x, M)^* A(T^{-N_k}x, M)\right)^{1/2M}, \quad k = 1, \ldots, n.$$

2. Compute all the orthonormal eigenspaces $U_i^{(M)}(T^{-N_k}x)$, $i = 1, \ldots, j-1$ of (11) (replacing N with N_k in (11)) and the eigenspace $U_j^{(M)}(T^{-N_1}x)$.

3. Let $\mathrm{proj}_V : \mathbb{R}^d \to \mathbb{R}^d$ be the orthogonal projection onto the subspace V so that $\ker(\mathrm{proj}_V) = V^\perp$ and $V_j^{(M)}(x) = \left(U_1^{(M)}(x) \oplus \cdots \oplus U_{j-1}^{(M)}(x)\right)^\perp$. Define $W_j^{(M,N_1)}(T^{-N_1}x) = U_j^{(M)}(T^{-N_1}x)$, and then define iteratively by pushing forward and taking orthogonal projections:

$$W_j^{(M,N_{k+1})}\left(T^{-N_{k+1}}x\right) = \mathrm{proj}_{V_j^{(M)}\left(T^{-N_{k+1}}x\right)}(A(T^{-N_k}x, N_{k+1} - N_k)W_j^{(M,N_k)}(T^{-N_k}x)).$$

4. $W_j^{(M,N_n)}(x) = W_j^{(M,0)}(x)$ is our approximation of $W_j(x)$.

Listing 2 shows an example implementation of Algorithm 2.2 in MATLAB. Lines 1–18 are responsible for performing Steps 1 and 2, whilst the push forward procedure of Step 3 is performed in lines 20–30. Again, the matrix cocycle is stored in $\mathtt{A} = \left[A(T^{-N}x) \mid A(T^{N-1}x) \mid \cdots \mid A(T^{M-1}x)\right]$ and the function returns $\mathtt{Wj}$ as its approximation of $W_j(x)$. The variable $\mathtt{Nk}$ is a one-dimensional array containing the elements of $\{N_k\}$ and is counted by $\mathtt{k}$.

Listing 2: Sample MATLAB code of Algorithm 2.2 to approximate $W_j(x)$.

```
1    k=0;
2    FOR n = Nk,
3      Psi = EYE(dim);
4      FOR i=0:M−1,
5        Psi = A(:,(n + i − 1)*dim + 1:(n + i)*dim)*Psi;
6        Psi = Psi/NORMEST(Psi);
7      END
8      [~,s,u] = SVD(Psi);
9      [~,p] = SORT(DIAG(s),'descend');
10     IF n==1,
11       Wj = u(:,p(j))/NORM(u(:,p(j)));
12     ELSE
13       FOR i = 1:j−1,
14         k = k+1;
15         U(:,i,k) = u(:,p(i))/NORM(u(:,p(i)));
16       END
17     END
18   END
19   k=0;
20   FOR n = 1:N,
21     Wj = A(:,(n−1)*dim+1:n*dim)*Wj;
22     Wj = Wj/NORM(Wj);
23     IF ANY(Nk == n+1),
24       k = k+1;
25       FOR i = 1:j−1,
26         Wj = Wj − DOT(Wj,U(:,i,k))*U(:,i,k);
27         Wj = Wj/NORM(Wj);
28       END
29     END
30   END
```

Remark 1. Unfortunately, some numerical issues with this approach remain. They stem primarily from the long multiplication involved in building the variable $\mathtt{Psi}$ of Listings 1 and 2. This results in $\mathtt{Psi}$ becoming too singular and hence $U_j^{(M)}(T^{-n}x)$ $(j \neq 1)$ poorly approximates $U_j(T^{-n}x)$. As can be seen in Section 5, Algorithm 2.2 works superbly for $W_2(x)$ as $U_1^{(M)}(T^{-n}x)$ is well approximated for large n. However when estimating $W_j(x), j > 2$, a good estimate of $U_{j-1}^{(M)}(x)$ is required for an accurate projection $\mathrm{proj}_{V_j^{(M)}}$ and for large n such an estimate becomes unreliable.

3. A dichotomy projector approach

We derive an approach for the computation of a vector $w_n^j \in W_n^j = \mathcal{R}(P_n^{j,s}) \cap \mathcal{R}(P_n^{j+1,u})$. For this task, we first need a guess of $\Lambda^{\mathrm{right}} \in R_i$ and of $\Lambda^{\mathrm{left}} \in R_{i+1}$ in two neighboring resolvent intervals that lie close to the common spectral interval, see Fig. 3.

Numerical experiments indicate that we get the best results by choosing Λ^{right} and Λ^{left} close to (but outside) the second Sacker–Sell interval. This conclusion is supported by theoretical estimates on the approximation error for Algorithm 3.2, discussed at the end of Section 3.

The following observation from [35,38] allows the computation of dichotomy projectors by solving

$$w_{n+1}^i = A_n w_n^i + \delta_{n,m-1}e_i, \quad n \in \mathbb{Z}, \ e_i \ i\text{-th unit vector.} \tag{12}$$
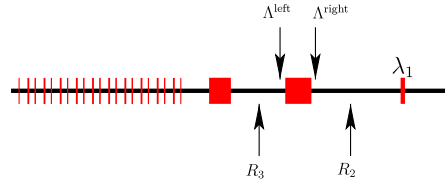
Fig. 3. Choice of $\Lambda^{\text{left/right}}$ in case $i = 2$.

With Green's function, cf. [27], the unique bounded solution $w_{\mathbb{Z}}^i$ of (12) has the explicit form

$$w_n^i = G(n, m)e_i, \quad n \in \mathbb{Z}, \text{ where } G(n, m) = \begin{cases} \Phi(n, m)P_m^s, & n \geq m, \\ -\Phi(n, m)P_m^u, & n < m, \end{cases} \tag{13}$$

and consequently

$$P_m^s = \begin{pmatrix} | & & | \\ w_m^1 & \cdots & w_m^d \\ | & & | \end{pmatrix}.$$

Numerically, we approximate the unique bounded solution on $\mathbb{Z}$ by the least squares solution of (12) on a sufficiently long interval. For an error analysis of this approximation process, we refer to [38, Theorem 4].

The algorithms that we propose in this section compute a vector $w \in W_0^j$ in analogy to $W_j(x)$ in the previous sections. For simplicity, we restrict the representation to the case $j = 2$ and assume that W_n^1 and W_n^2 are one-dimensional subspaces.

In the absence of information about the dichotomy intervals, one may proceed as follows. Given a finite sequence of matrices, one can estimate a point in the spectral interval $[\lambda_q^-, \lambda_q^+]$, $q = 1, 2, 3$ by computing the (logarithmic) growth rates of one-, two-, and three-dimensional subspaces using direct multiplication; these growth rates should approximate λ_1, $\lambda_1 + \lambda_2$, and $\lambda_1 + \lambda_2 + \lambda_3$, respectively. By taking differences to obtain estimates $\hat{\lambda}_q$, $q = 1, 2, 3$, (the caret indicating estimated quantities) one should obtain values in the interior of $[\lambda_q^-, \lambda_q^+]$, $q = 1, 2, 3$. We then estimate $\Lambda^{\text{left}} = \hat{\lambda}_2 - (\hat{\lambda}_2 - \hat{\lambda}_3)/10 \lessapprox \lambda_2^-$ and $\Lambda^{\text{right}} = \hat{\lambda}_2 - (\hat{\lambda}_2 - \hat{\lambda}_1)/10 \gtrapprox \lambda_2^+$.

In the first step of our first algorithm, we compute a basis of the two-dimensional space $\mathcal{R}(P_0^{3,u})$. Then, in the second step, we search for the direction w in this subspace that additionally lies in $\mathcal{R}(P_0^{2,s})$ and assure in this way that $w \in \mathcal{R}(P_0^{2,s}) \cap \mathcal{R}(P_0^{3,u}) = W_0^2$.

Algorithm 3.1 (*A Dichotomy Projector Approach to Estimate $W_2(x)$ by Computing W_0^2*).

1. Suppose $N \in \mathbb{N}$ and consider $n \in [-N, N] \cap \mathbb{Z}$. Let $A_n = A(T^n x)$.
   Solve the least squares problem

   $$\tilde{w}_{n+1}^i = e^{-\Lambda^{\text{left}}} A_n \tilde{w}_n^i + \delta_{n,-1} r^i, \quad n = -N, \ldots, N-1, i = 1, 2 \text{ such that } \|(\tilde{w}_{-N}^i, \ldots, \tilde{w}_N^i)\|_2 \text{ is minimized}, \tag{14}$$

   where the r^i are chosen at random and $\|\cdot\|_2$ is the ℓ^2-norm. Define $p^i := A_{-1} \tilde{w}_{-1}^i$, $i = 1, 2$.

2. Solve for $\tilde{w}_{[0,N]}$ and κ the least squares problem

   $$\tilde{w}_{n+1} = e^{-\Lambda^{\text{right}}} A_n \tilde{w}_n, \quad n = 0, \ldots, N-1, \tag{15}$$

   $$\tilde{w}_0 + \kappa p^1 + p^2 = 0, \quad \text{such that } \|(\tilde{w}_0, \ldots, \tilde{w}_N, \kappa)\|_2 \text{ is minimized}. \tag{16}$$

   Then $\tilde{w}_0$ is our approximation of $w_2(x) \in W_2(x)$.

The unique bounded solutions on $\mathbb{Z}$ of these two steps satisfy $p^1, p^2 \in \mathcal{R}(P_0^{3,u})$ and these vectors are generically linear independent. Furthermore $\tilde{w}_0 \in \mathcal{R}(P_0^{2,s})$ due to (15) and $\tilde{w}_0 \in \mathcal{R}(P_0^{3,u})$ due to (16). Thus $\tilde{w}_0 \in \mathcal{R}(P_0^{3,u}) \cap \mathcal{R}(P_0^{2,s}) = W_0^2$.

Note that (14) has the form

$$B\tilde{w} = r, \quad \text{with } B \in M_{2dN, d(2N+1)}(\mathbb{R}), \ r \in \mathbb{R}^{2dN},$$

where

$$B = \begin{pmatrix} -e^{-\Lambda^{\text{left}}} A_{-N} & I & & \\ & \ddots & \ddots & \\ & & -e^{-\Lambda^{\text{left}}} A_N & I \end{pmatrix}, \qquad \tilde{w} = \begin{pmatrix} \tilde{w}_{-N} \\ \vdots \\ \tilde{w}_{N-1} \end{pmatrix},$$

and the nth entry of r is the vector $\delta_{n,-1} r^i \in \mathbb{R}^d$ for $i = 1, 2$.

The least squares solution can be obtained, using the Moore–Penrose inverse:

$$\tilde{w} = B^+ r, \quad \text{where } B^+ = B^T (BB^T)^{-1},$$

cf. [39]. Numerically, we find $\tilde{w}$ by solving the linear system $BB^T y = r$; then $\tilde{w} = B^T y$.

Listing 3: Sample MATLAB code for Algorithm 3.1.

```matlab
1   % step 1
2   B = ZEROS(2*N*dim,2*(N+1)*dim);
3   FOR i = 1:2*N
4     B(dim*(i-1)+1:dim*i,dim*(i-1)+1:dim*i)
5       = -EXP(-Lambda_left)*A(:,dim*(i-1)+1:dim*i);
6     B(dim*(i-1)+1:dim*i,dim*(i)+1:dim*(i+1))
7       = EYE(dim);
8   END
9   R = ZEROS(2*dim*N,2);
10  R(dim*(N-1)+1:dim*N,:) = RAND(dim,2);
11  y = (B*B')\R;
12  u = B'*y;
13  p1 = A(:,dim*(N-1)+1:dim*N)*u(dim*(N-1)+1:dim*N,1);
14  p2 = A(:,dim*(N-1)+1:dim*N)*u(dim*(N-1)+1:dim*N,2);
15  p1 = v1/NORM(v1); v2 = v2/NORM(v2);
16  % step 2
17  B = ZEROS(dim*(N+1),dim*(N+1)+1);
18  FOR i = 0:N-1
19    B(dim*i+1:dim*(i+1),dim*i+1:dim*(i+1))
20      = -EXP(-Lambda_right)*A(:,dim*(i+N)+1:dim*(i+N+1));
21    B(dim*i+1:dim*(i+1),dim*(i+1)+1:dim*(i+2))
22      = EYE(dim);
23  END
24  B(dim*N+1:dim*(N+1),1:dim) = EYE(dim);
25  B(dim*N+1:dim*(N+1),dim*(N+1)+1) = p1;
26  R = ZEROS(dim*(N+1),1);
27  R(dim*N+1:dim*(N+1),1) = -p2;
28  y = (B*B')\R;
29  u = B'*y;
30  w2 = u(dim*N+1:dim*(N+1))/NORM(u(dim*N+1:dim*(N+1)));
```

Note that in the unlikely case where $p^1 \in W_0^2$, Algorithm 3.1 fails. An alternative approach for computing vectors in W_0^2 that avoids this problem is introduced in Algorithm 3.2. The main idea of this algorithm is to take a random vector r, project it first to $\mathcal{R}(P_0^{3,u})$ and then eliminate components in the wrong subspaces, by projecting with $P_0^{2,s}$.

Algorithm 3.2 (*An Alternate Dichotomy Projector Approach*)**.**

1. Again, suppose $N \in \mathbb{N}$ and consider $n \in [-N, N] \cap \mathbb{Z}$ and let $A_n = A(T^n x)$ as above. Solve the least squares problem

$$\tilde{w}_{n+1} = e^{-\Lambda^{\text{left}}} A_n \tilde{w}_n + \delta_{n,-1} r, \quad n = -N, \ldots, N-1, \text{ such that } \|(\tilde{w}_{-N}, \ldots, \tilde{w}_N)\|_2 \text{ is minimized,} \tag{17}$$

   where r is chosen at random, and define $r' = A_{-1}\tilde{w}_{-1}$.
2. Solve the least squares problem

$$\tilde{w}'_{n+1} = e^{-\Lambda^{\text{right}}} A_n \tilde{w}'_n + \delta_{n,-1} r', \quad n = -N, \ldots, N-1, \text{ such that } \|(\tilde{w}'_{-N}, \ldots, \tilde{w}'_N)\|_2 \text{ is minimized.} \tag{18}$$

   Then $\tilde{w}'_0$ is our approximation of $w_2(x) \in W_2(x)$.

The solution w_0 on $\mathbb{Z}$ of these two steps satisfies $w_0 = P_0^{2,s} P_0^{3,u} r \in \mathcal{R}(P_0^{2,s}) \cap \mathcal{R}(P_0^{3,u}) = W_0^2$.

Listing 4: Sample MATLAB code for the second step of Algorithm 3.2.

```matlab
1   B = ZEROS(2*N*dim,2*(N+1)*dim);
2   FOR i = 1:2*N
3     B(dim*(i-1)+1:dim*i,dim*(i-1)+1:dim*i)
4       = -EXP(-Lambda_right)*A(:,dim*(i-1)+1:dim*i);
5     B(dim*(i-1)+1:dim*i,dim*i+1:dim*(i+1)) = EYE(dim);
6   END
7   R = ZEROS(2*dim*N,1);
8   R(dim*(N-1)+1:dim*N,:) = p1;
9   y = (B*B')\R;
10  u = B'*y;
11  w2 = u(dim*N+1:dim*(N+1))/NORM(u(dim*N+1:dim*(N+1)));
```

3.1. Error estimate

We give an error estimate for the solution of Algorithm 3.2 for a finite choice of N. Details on deriving this estimate are postponed to a forthcoming publication.

For Λ^{left} and Λ^{right} close to the boundary of the second Sacker–Sell spectral interval, we denote the dichotomy rates of

$$w_{n+1} = e^{-\Lambda^{\text{left}}} A_n w_n, \qquad w_{n+1} = e^{-\Lambda^{\text{right}}} A_n w_n, \quad n \in \mathbb{Z} \tag{19}$$

by $(\alpha^{\ell,s}, \alpha^{\ell,u})$ and $(\alpha^{r,s}, \alpha^{r,u})$, respectively. Let w_0 be the solution of Algorithm 3.2 on $\mathbb{Z}$ and let $\tilde{w}_0$ be its approximation for a finite choice of N. Careful estimates show that the approximation error in the "wrong subspace" $\mathcal{R}(Q)$, with $Q := I - P_0^{2,s} P_0^{3,u}$ is given as

$$\|Q(w_0 - \tilde{w}_0)\| \leq CN(e^{-\alpha^{\ell,s}N} + e^{-\alpha^{r,u}N}), \tag{20}$$

where the constant $C > 0$ does not depend on N.

The exponential dichotomy rates $\alpha^{\ell,s}$ and $\alpha^{r,u}$ of the difference equations (19) depend on the choice of Λ^{left} and Λ^{right} in the following way: for Λ^{left} in the resolvent set $R_3 = [\lambda_3^+, \lambda_2^-]$ the difference equation

$$w_{n+1} = e^{-\Lambda^{\text{left}}} A_n w_n, \quad n \in \mathbb{Z}$$

has an exponential dichotomy with stable dichotomy rate $\alpha^{\ell,s}$ for all $\alpha^{\ell,s}$ with

$$0 < \alpha^{\ell,s} < \Lambda^{\text{left}} - \lambda_3^+.$$

Similarly, for Λ^{right} in the resolvent set $R_2 = [\lambda_2^+, \lambda_1^-]$ the difference equation

$$w_{n+1} = e^{-\Lambda^{\text{right}}} A_n w_n, \quad n \in \mathbb{Z}$$

has an exponential dichotomy with unstable dichotomy rate $\alpha^{r,u}$ for all $\alpha^{r,u}$ with

$$0 < \alpha^{r,u} < \lambda_1^- - \Lambda^{\text{right}}.$$

Note that both of the above inequalities are strict.

Inspecting Eq. (20), we get the best (smallest) maximal error if we choose $\Lambda^{\text{left}} \in R_3$ and $\Lambda^{\text{right}} \in R^2$ so as to maximize $\alpha^{\ell,s}$ and $\alpha^{r,u}$. Consequently, we get the best numerical approximations, if Λ^{left} and Λ^{right} are chosen close to, but not equal to, the boundary of the common spectral interval $[\lambda_2^-, \lambda_2^+]$.

4. The Ginelli and Wolfe schemes

4.1. The Ginelli scheme

The Ginelli Scheme was first presented by Ginelli et al. in [36] as a method for accurately computing the covariant Lyapunov vectors of an orbit of an invertible differentiable dynamical system where the $A(x) = DT(x)$ are the Jacobian matrices of the flow or map.

Estimates of the $W_j(x)$ are found by constructing equivariant subspaces $S_j(x) = W_1(x) \oplus \cdots \oplus W_j(x)$ and filtering the invariant directions contained therein using a power method on the inverse system restricted to the subspaces $S_j(x)$.

To construct the subspaces $S_j(x)$ we utilize the notion of the stationary Lyapunov basis [24]. Choose j orthonormal vectors $s_1(T^{-n}x), s_2(T^{-n}x), \ldots, s_j(T^{-n}x), n \geq 1$, such that $s_i(T^{-n}x) \notin V_{j+1}(T^{-n}x)$ for $1 \leq i \leq j$ and construct

$$\tilde{s}_i^{(n)}(x) = A(T^{-n}x, n)s_i(T^{-n}x), \quad i = 1, \ldots, j.$$

Using the Gram–Schmidt procedure, construct the orthonormal basis $\{s_1^{(n)}(x), \ldots, s_j^{(n)}(x)\}$ from $\{\tilde{s}_1^{(n)}(x), \ldots, \tilde{s}_j^{(n)}(x)\}$, that is,

$$s_1^{(n)}(x) = \frac{1}{\left\| \tilde{s}_1^{(n)}(x) \right\|} \tilde{s}_1^{(n)}(x),$$

$$s_2^{(n)}(x) = \frac{1}{\left\| \left(\tilde{s}_2^{(n)}(x) - \left(\tilde{s}_2^{(n)}(x) \cdot s_1^{(n)}(x) \right) s_1^{(n)}(x) \right) \right\|} \left(\tilde{s}_2^{(n)}(x) - \left(\tilde{s}_2^{(n)}(x) \cdot s_1^{(n)}(x) \right) s_1^{(n)}(x) \right), \tag{21}$$

$$\vdots$$

Then as $n \to \infty$ the basis $\{s_1^{(n)}(x), \ldots, s_j^{(n)}(x)\}$ converges to a set of orthonormal vectors $\{s_1^{(\infty)}(x), \ldots, s_j^{(\infty)}(x)\}$ which span the j fastest expanding directions of the cocycle A [24], that is, if the multiplicities $m_1 = \cdots = m_j = 1$

$$S_j(x) := \text{span} \left\{ s_1^{(\infty)}(x), \ldots, s_j^{(\infty)}(x) \right\} = W_1(x) \oplus \cdots \oplus W_j(x)$$

$$= V_1(x) \setminus V_{j+1}(x). \tag{22}$$

If the Oseledets subspaces are not all one-dimensional, that is the Lyapunov spectrum is degenerate, then we choose $S_j(x)$ only for those j which are the sum of the first k multiplicities, i.e., $j = m_1 + \cdots + m_k$. Then

$$S_j(x) = W_1(x) \oplus \cdots \oplus W_k(x)$$

$$= V_1(x) \setminus V_{k+1}(x).$$

In the interest of readability we assume the Oseledets subspaces are one-dimensional but note that the approach may be extended to the multi-dimensional case.

Note that the $S_j(x)$ are equivariant by construction:

$$A(x, n)S_j(x) = S_j(T^n x)$$

provided $j \leq m_1 + \cdots + m_{\ell-1}$ if $\lambda_\ell = -\infty$.

We describe the Ginelli approach to finding $W_2(x)$. Suppose $\dim W_1(x) = \dim W_2(x) = 1$ and $\lambda_1 > \lambda_2 > -\infty$ and that the basis $\{s_1^{(\infty)}(x), s_2^{(\infty)}(x)\}$ is known at $x \in X$. Note first that $\operatorname{span}\left\{s_1^{(\infty)}(x)\right\} = W_1(x)$. Let $c(x) \in \mathbb{R}^2$ denote the coefficients of $w_2(x) \in W_2(x)$ in the basis $\{s_1^{(\infty)}(x), s_2^{(\infty)}(x)\}$ (recall that the orthogonal projection of $w_2(x)$ onto $s_i^{(\infty)}(x)$ is zero for $i = 3, 4, \ldots, d$) then

$$w_2(x) = c_1(x)s_1^{(\infty)}(x) + c_2(x)s_2^{(\infty)}(x).$$

Lemma 4.1. *Let $Q(x)$ denote the $d \times 2$ matrix whose ith column is $s_i^{(\infty)}(x)$. Then for each $n \geq 0$ there exists an upper triangular, 2×2 matrix $R(x, n)$ satisfying*

$$A(x, n)Q(x) = Q(T^n x)R(x, n). \tag{23}$$

Proof. Note that

$$A(x, n)Q(x) = A(x, n) \begin{pmatrix} | & | \\ s_1^{(\infty)}(x) & s_2^{(\infty)}(x) \\ | & | \end{pmatrix}$$

$$= \begin{pmatrix} | & | \\ A(x, n)s_1^{(\infty)}(x) & A(x, n)s_2^{(\infty)}(x) \\ | & | \end{pmatrix}$$

$$= Q(T^n x)R(x, n),$$

where

$$Q(T^n x) = \begin{pmatrix} | & | \\ s_1^{(\infty)}(T^n x) & s_2^{(\infty)}(T^n x) \\ | & | \end{pmatrix}$$

and

$$R(x, n) = \begin{pmatrix} \left\| A(x, n)s_1^{(\infty)}(x) \right\| & \langle s_1^{(\infty)}(T^n x), A(x, n)s_2^{(\infty)}(x) \rangle \\ 0 & \left\| A(x, n)s_2^{(\infty)}(x) \right\| \end{pmatrix}, \tag{24}$$

using the equivariance of $S_1(x) = \operatorname{span}\{s_1^{(\infty)}(x)\}$ and $S_2(x) = \operatorname{span}\{s_1^{(\infty)}(x), s_2^{(\infty)}(x)\}$. $\square$

Thus, the QR-decomposition of Lemma 4.1 is equivalent to the Gram–Schmidt orthonormalization that defines the stationary Lyapunov bases. The columns of $Q(T^n x)$ form the stationary Lyapunov basis at $T^n x$.

We have chosen the above notation $R(x, n)$ specifically since, defined in this way, R forms a cocycle which is the restriction of A to the invariant subspaces S_j.

Lemma 4.2. *The matrix $R(x, n)$ defined above forms a cocycle over T.*

Proof. Let $n, m \geq 0$ then

$$A(x, n + m)Q(x) = Q(T^{n+m}x)R(x, n + m) \tag{25}$$

by Lemma 4.1. Since $A(x, n + m) = A(T^n x, m)A(x, n)$,

$$A(x, n + m)Q(x) = A(T^n x, m)A(x, n)Q(x)$$
$$= A(T^n x, m)Q(T^n x)R(x, n)$$
$$= Q(T^{n+m}x)R(T^n x, m)R(x, n). \tag{26}$$

Equating (25) and (26) gives

$$R(T^n x, m)R(x, n) = R(x, n + m),$$

as $Q(T^{n+m}x)$ is left-invertible. $\square$

Since $c(x)$ is the vector of coefficients of the second Oseledets vector of the cocycle A, it is the second Oseledets vector of the cocycle R. To see this, recall $w_2(x) = Q(x)c(x) \in W_2(x)$ so that

$$\lambda_2 = \lim_{n \to \infty} \frac{1}{n} \log \|A(x, n)Q(x)c(x)\|$$

which, due to (23), becomes

$$
\lambda_2 = \lim_{n \to \infty} \frac{1}{n} \log \left\| Q(T^n x) R(x, n) c(x) \right\|
$$

$$
= \lim_{n \to \infty} \frac{1}{n} \log \| R(x, n) c(x) \|
$$

since the columns of $Q(T^n x)$ are orthonormal.

We may approximate $c(x)$ numerically using a simple power method on the inverse cocycle R^{-1} (which exists since $\lambda_1 > \lambda_2 > -\infty$). The Ginelli method can be summarized by the following steps:

Algorithm 4.3 (*Ginelli Method of Approximating $w_2(x) \in W_2(x)$*)**.**

1. Choose $x \in X$ and $M > 0$ and form $\{s_1^{(M)}(x), s_2^{(M)}(x)\}$ by first randomly selecting two orthonormal vectors $\{s_1(T^{-M}x), s_2(T^{-M}x)\}$ then performing the push-forward/Gram–Schmidt procedure given by (21). That is, define

$$
\tilde{s}_i^{(M)}(x) = A(T^{-M}x, M)s_i(T^{-M}x), \quad i = 1, 2,
$$

followed by setting

$$
s_1^{(M)}(x) = \mathcal{N}\left(\tilde{s}_1^{(M)}(x)\right),
$$

$$
s_2^{(M)}(x) = \mathcal{N}\left(\tilde{s}_2^{(M)}(x) - \left(\tilde{s}_2^{(M)}(x) \cdot s_1^{(M)}(x)\right) s_1^{(M)}(x)\right),
$$

where $\mathcal{N} : v \mapsto v/\|v\|$. The vectors $\{s_1^{(M)}(x), s_2^{(M)}(x)\}$ form an approximation to the stationary Lyapunov basis $\{s_1^{(\infty)}(x), s_2^{(\infty)}(x)\}$.

2. Choose $N > 0$ and using the approximate basis $\{s_1^{(M)}(x), s_2^{(M)}(x)\}$ in (24), form an approximation to $R(x, N)$, denoted by $R^{(M)}(x, N)$.

3. Choose $c' \in \mathbb{R}^2$ either at random, or by some guess at the second Oseledets vector of R at $T^N(x) \in X$, in this review we found $c' = (0, 1)$ to work well. Use the inverse iteration method to approximate $c(x)$, that is, define our approximation to $c(x)$ as

$$
c^{(M,N)}(x) = R^{(M)}(x, N)^{-1} c'
$$

$$
= R^{(M)}(T^N x, -N) c'.
$$

4. Then

$$
w_2^{(M,N)}(x) = \begin{pmatrix} | & | \\ s_1^{(M)}(x) & s_2^{(M)}(x) \\ | & | \end{pmatrix} c^{(M,N)}(x)
$$

is our approximation to $w_2(x) \in W_2(x)$.

As before, there is some freedom of choice of both M and N as well as of the initial orthonormal basis $\{s_1(T^{-M}x), s_2(T^{-M}x)\}$, used to approximate $S_2(x)$, and of the 2-tuple c'. The larger M and N are chosen, the more accurate $w_2^{(M,N)}(x)$ will be, provided $s_2(T^{-M}x) \notin W_1(T^{-M}x) \cup V_3(T^{-M}x)$ and $c' \notin E_1(T^M x)$ where E_1 is the Oseledets subspace of R with Lyapunov exponent λ_1.

Listing 5: Sample MATLAB code for Algorithm 4.3.

```
1   [Q,~] = QR(RAND(dim,j),0);c = [ZEROS(1,j−1) 1];
2   FOR i = 1:N,
3     QNew = A(:,(i−1)*dim+1:i*dim)*Q;
4     [Q,~] = QR(QNew,0);
5   END
6   Q0 = Q;
7   FOR i = N+1:2*N+1,
8     QNew = A(:,(i−1)*dim +1: i*dim)*Q;
9     [Q,R] = QR(QNew,0);
10    AllR = HORZCAT(R,AllR);
11  END
12  numOfR = SIZE(AllR,2)/j;
13  FOR i = 1:numOfR
14    R = AllR(:,(i−1)*j+1:i*j);
15    cNew = R\c;
16    c = cNew/NORM(cNew);
17  END
18  w = Q0*c;
```

Listing 5 shows an example implementation of Algorithm 4.3 in MATLAB which approximates $w_j(x) \in W_j(x)$ using $M = N$ and $s_1(T^{-M}x), \ldots, s_j(T^{-M}x)$ are chosen at random and $c' = \underbrace{(0, \ldots, 0, 1)}_{j-1 \text{ entries}}$. Lines 2 through 6 construct the approximation of the

stationary Lyapunov basis, $\left\{s_1^{(M)}(x), \ldots, s_j^{(M)}(x)\right\}$ which is stored as columns of the matrix $Q(x)$ represented as the variable Q0, while lines 7 through 11 construct the cocycle R stored in AllR as $[R(x) \mid R(Tx)\mid\cdots\mid R(T^N x)]$. Lines 12 through 17 perform a simple power method on $R(x, N)^{-1}$ to find the coefficient vector c, which represents the approximation of $w_j(x)$ in the basis $\left\{s_1^{(M)}(x), \ldots, s_j^{(M)}(x)\right\}$. Thus, the approximation is given by $Q(x)c$. Although Algorithm 4.3 is specific to the case where $j = 2$, Listing 5 is applicable to any j for which $R(x, N)^{-1}$ exists.

It can be shown that, in this case where the top Lyapunov exponent has multiplicity 1, $E_1 = \text{span}\{(1, 0)^T\}$.

Lemma 4.4. *If the first Lyapunov exponent has multiplicity* 1, *the dominant Oseledets subspace of the cocycle R is $E_1 = \text{span}\{(1, 0)^T\}$.*

Proof. Recall that $s_1^{(\infty)}(x) \in W_1(x)$ since span $\left\{s_1^{(\infty)}(x)\right\} = S_1(x) = V_1(x) \setminus V_2(x)$ (from (22)) and for all $s \in V_1(x) \setminus V_2(x)$

$$\lambda_1 = \lim_{n\to\infty} \frac{1}{n} \log \|A(x, n)s\| .$$

We may write $s = Q(x)(a, 0)^T$ for some $a \in \mathbb{R}$ then

$$\lambda_1 = \lim_{n\to\infty} \frac{1}{n} \log \left\|A(x, n)Q(x)(a, 0)^T\right\|$$

$$= \lim_{n\to\infty} \frac{1}{n} \log \left\|Q(T^n x)R(x, n)(a, 0)^T\right\|$$

and since $\|Q(T^n x)s\| = \|s\|$ (the columns of Q are orthonormal)

$$\lambda_1 = \lim_{n\to\infty} \frac{1}{n} \log \left\|R(x, n)(a, 0)^T\right\| .$$

Since S_1 is A-invariant, span$\{(1, 0)^T\}$ is R-invariant and the proof is complete. $\square$

4.1.1. Limited data scenario

In the case where convergence is not satisfactory because the amount of cocycle data available is too small (for any M and N to be small), the approximations from Algorithm 4.3 can be improved by using better guesses at $s_1(T^{-M}x)$, $s_2(T^{-M}x)$ and c'.

Note that $s_1^{(\infty)}(x)$ and $s_2^{(\infty)}(x)$ are two orthonormal vectors optimized for maximal growth over the time interval $[-\infty, 0] \cap \mathbb{Z}$. In the situation where the values of M and N are limited, one can choose those two vectors that are optimized for growth over the shorter time interval $[-M, 0] \cap \mathbb{Z}$. In [37] this is achieved by computing the left singular vectors of $A(T^{-M}x, M)$. This approach works well for small M but can become inaccurate for very large M for the reasons in Remark 1.

In practice, we have observed that a combination of Step 1 in Algorithm 4.3 and the above provides the most robust method of accurately approximating $s_1^{(\infty)}(x)$ and $s_2^{(\infty)}(x)$. As an alternative to Algorithm 4.3 the following may be used: For $M \geq M'$, compute vectors optimized for growth from $-M$ to $-M + M'$, then push-forward these vectors from $-M + M'$ to 0.

Algorithm 4.5 (*Improved* Algorithm 4.3).

1. Choose $x \in X$ and $M \geq M' > 0$. Compute the two left singular vectors of $A(T^{-M}x, M')$ corresponding to the two largest singular values and call them $\tilde{s}_1(T^{-M+M'}x)$ and $\tilde{s}_2(T^{-M+M'}x)$. Now define $\left\{s_1^{(M,M')}(x), s_2^{(M,M')}(x)\right\}$ as an approximation to $\left\{s_1^{(\infty)}(x), s_2^{(\infty)}(x)\right\}$ by the Gram–Schmidt orthonormalization of $A(T^{-M+M'}x, M - M')s_1(T^{-M+M'}x)$ and $A(T^{-M+M'}x, M - M')s_2(T^{-M+M'}x)$ as in (21).

Steps 2–4 as in Algorithm 4.3.

In practice, one should choose M' large enough so that enough data is sampled, but not so large that $A(T^{-M}x, M')$ is too singular.

4.2. The Wolfe scheme

The approach followed by Wolfe et al. [37] directly computes the subspace splitting as the intersection of two sets of invariant subspaces. The description of the numerical construction of the subspaces $S_j(x)$ featured below differs slightly from [37], however, the essential features of the approach are retained. In fact, the constructions featured here improve upon those in [37] in terms of accuracy versus amount of cocycle data used—in the notation of Algorithm 4.6, if M_1 is made larger, $w_2^{(M_1, M_1', M_2)}(x)$ is more accurate, which is not the case in [37] for the same reasons discussed in Remark 1.

Recall the eigenspace decomposition $U_j(x)$ of the limiting matrix $\Psi(x)$ presented in Section 2 and define $V_j(x) = U_j(x) \oplus \cdots \oplus U_\ell(x)$. Recall that $V_j(x) \supset W_j(x), W_{j+1}(x), \ldots, W_\ell(x)$. Also recall from the previous section that $S_j(x) \supset W_1(x), W_2(x), \ldots, W_j(x)$. Thus

$$W_j(x) = V_j(x) \cap S_j(x).$$

Again, in the interest of readability we assume the Oseledets subspaces $W_j(x)$, and the eigenspaces $U_j(x)$, are one-dimensional. As in the case of the previous section, the ideas here may be extended to the case in which the Oseledets subspaces are not one-dimensional.

Let $u_j(x)$ be the singular vector spanning $U_j(x)$ and let $s_j(x) = s_j^{(\infty)}(x)$ be the jth element of the stationary Lyapunov basis as in the previous section. Then note

$$w_j(x) = \sum_{i=1}^{j} \langle w_j(x), s_i(x) \rangle s_i(x),$$

$$w_j(x) = \sum_{i=j}^{d} \langle w_j(x), u_i(x) \rangle u_i(x).$$

Taking inner products with $u_k(x)$ and $s_k(x)$ respectively gives

$$\langle w_j(x), u_k(x) \rangle = \sum_{i=1}^{j} \langle w_j(x), s_i(x) \rangle \langle s_i(x), u_k(x) \rangle \quad \text{for } k \geq j, \tag{27}$$

$$\langle w_j(x), s_k(x) \rangle = \sum_{i=j}^{d} \langle w_j(x), u_i(x) \rangle \langle u_i(x), s_k(x) \rangle \quad \text{for } k \leq j. \tag{28}$$

Substituting (27) into (28) and rearranging gives

$$\langle w_j(x), s_k(x) \rangle = \sum_{i=1}^{j} \left(\sum_{h=j}^{d} \langle s_k(x), u_h(x) \rangle \langle u_h(x), s_i(x) \rangle \right) \langle s_i(x), w_j(x) \rangle. \tag{29}$$

Note that $\sum_{h=1}^{d} \langle s_k(x), u_h(x) \rangle \langle u_h(x), s_i(x) \rangle = \delta_{ki}$ so

$$\sum_{h=j}^{d} \langle s_k(x), u_h(x) \rangle \langle u_h(x), s_i(x) \rangle = \delta_{ki} - \sum_{h=1}^{j-1} \langle s_k(x), u_h(x) \rangle \langle u_h(x), s_i(x) \rangle.$$

Then (29) becomes

$$
\begin{aligned}
\langle w_j(x), s_k(x) \rangle &= \sum_{i=1}^{j} \delta_{ki} \langle s_i(x), w_j(x) \rangle - \sum_{i=1}^{j} \sum_{h=1}^{j-1} \langle s_k(x), u_h(x) \rangle \langle u_h(x), s_i(x) \rangle \langle s_i(x), w_j(x) \rangle \\
&= \langle s_k(x), w_j(x) \rangle - \sum_{i=1}^{j} \sum_{h=1}^{j-1} \langle s_k(x), u_h(x) \rangle \langle u_h(x), s_i(x) \rangle \langle s_i(x), w_j(x) \rangle,
\end{aligned}
$$

$$0 = \sum_{i=1}^{j} \sum_{h=1}^{j-1} \langle s_k(x), u_h(x) \rangle \langle u_h(x), s_i(x) \rangle \langle s_i(x), w_j(x) \rangle. \tag{30}$$

Eq. (30) may be considered as a $j \times j$ homogeneous linear equation by defining a matrix entry-wise as

$$D_{ki} = \sum_{h=1}^{j-1} \langle s_k(x), u_h(x) \rangle \langle u_h(x), s_i(x) \rangle$$

and solving

$$Dy = 0,$$

where $y_i = \langle s_i(x), w_j(x) \rangle$. The entries of y are then the coefficients of $w_j(x)$ with respect to the basis $s_1(x) \ldots, s_j(x)$.

The Wolfe approach may be implemented as follows:

Algorithm 4.6 (*Improved Wolfe Approach to Approximating $w_j(x) \in W_j(x)$*).

1. Choose $x \in X$ and $M_1 \geq M_1' > 0$ and construct $\{s_1^{(M_1, M_1')}(x), \ldots, s_j^{(M_1, M_1')}(x)\}$ as an approximation of the stationary Lyapunov basis vectors $\{s_1^{(\infty)}(x), \ldots, s_j^{(\infty)}(x)\}$ using the methods outlined in Step 1 of Algorithm 4.5, that is, compute the left singular vectors of $A(T^{-M_1}x, M_1')$ corresponding to the $j-1$ largest singular values and call them $\tilde{s}_i(T^{-M_1+M_1'}x)$, $i = 1, \ldots, j-1$. Then form the $s_i^{(M_1, M_1')}(x)$ by the Gram–Schmidt orthonormalization of $A(T^{-M_1+M_1'}x, M_1 - M_1')\tilde{s}_i(T^{-M_1+M_1'}x)$ for $i = 1, \ldots, j-1$.

2. Choose $M_2 > 0$ and construct the one-dimensional eigenspaces $U_1^{(M_2)}(x), \ldots, U_{j-1}^{(M_2)}(x)$ as approximations to the eigenspaces $U_1(x), \ldots, U_{j-1}(x)$ as in Step 1 of Algorithm 2.1, that is, construct

$$\Psi^{(M_2)}(x) = \left(A(x, M_2)^* A(x, M_2) \right)^{1/2M_2},$$

and let $U_i^{(M_2)}(x)$ be the ith orthonormal eigenspace of $\Psi^{(M_2)}(x)$. Define $u_i^{(M_2)}(x) \in U_i^{(M_2)}(x)$, $i = 1, \ldots, j-1$.

3. Form the matrix D as above:

$$D_{ki} = \sum_{h=1}^{j-1} \langle s_k^{(M_1, M_1')}(x), u_h^{(M_2)}(x) \rangle \langle u_h^{(M_2)}(x), s_i^{(M_1, M_1')}(x) \rangle.$$

4. Solve the homogeneous linear equation $Dy = 0$. Then $w_j^{(M_1, M_1', M_2)} = \sum_{i=1}^{j} y_i s_i(x)$ forms our approximation of $w_j(x) \in W_j(x)$.

This approach suffers from the same numerical stability issue of Algorithm 2.2 of Section 2.1. Namely, the vector spaces $U_1^{(M)}(x), \ldots, U_{j-1}^{(M)}(x)$ may only poorly approximate $U_1(x), \ldots, U_{j-1}(x)$ for M too large (see the final paragraph of Section 2.1).

A recent paper [40] provides alternative descriptions of both the Ginelli et al. and Wolfe and Samelson methods, well-suited to those familiar with the QR-decomposition based numerical method for estimating Lyapunov exponents due to Benettin et al. [17,18] and Shimada and Nagashima [19]. The discussion in [40] is restricted to invertible cocycles generated by the Jacobian matrices of a dynamical system. Although this assumption allows stable numerical methods to be constructed, i.e., better convergence obtained for larger data sets, it means some important examples in which the matrix cocycle is non-invertible are overlooked, for example the case study of Section 5.3. While the memory footprint of the implementations discussed in [40] is estimated, there is no discussion of convergence rates or accuracy with respect to the amount of cocycle data available. Finally, while the examples featured in [40] explain the methods presented in the context of differentiable dynamical systems the case studies of Section 5 in the present paper focus on comparative performance of the methods presented, via a broad range of possible applications.

5. Numerical comparisons of the four approaches

We present three detailed case studies, comparing the four approaches for calculating Oseledets subspaces. The first case study is a nontrivial model for which we know the Oseledets subspaces exactly and can therefore precisely measure the accuracy of the methods. The second case study produces a relatively low-dimensional matrix cocycle, while the third case study generates a very high-dimensional matrix cocycle; in these case studies we use two fundamental properties of Oseledets subspaces to assess the accuracy of the four approaches.

5.1. Case study 1: an exact model

In general the Oseledets subspaces cannot be found analytically which makes the task of determining the efficacy of the above approaches difficult. However, the exact model described below allows us to compare the numerical approximations with the exact solution by building a cocycle in which the subspaces are known *a priori*.

We generate a system with simple Lyapunov spectrum $\lambda_1 > \lambda_2 > \cdots > \lambda_d > -\infty$. We form a diagonal matrix

$$R = \begin{pmatrix} e^{\lambda_1} & & & 0 \\ & e^{\lambda_2} & & \\ & & \ddots & \\ 0 & & & e^{\lambda_d} \end{pmatrix}$$

and generate the cocycle by the sequence of matrices $\{A_n\}$ where

$$A_n = S_n R S_{n-1}^{-1}$$

$$S_n = \begin{cases} I + \epsilon Z, & \text{for } n \neq -1, \ n \in [-N, N] \cap \mathbb{Z}, \\ I + \begin{pmatrix} 0 & & & \\ z_2 & 0 & & \\ & \ddots & \ddots & \\ & & z_d & 0 \end{pmatrix}, & \text{for } n = -1. \end{cases} \tag{31}$$

The entries of Z and the numbers $z_2, \ldots, z_d$ are uniformly randomly generated from the interval $[0, 1]$. By construction, the columns of S_{n-1} span the Oseledets subspaces at time $n \in [-N, N] \cap \mathbb{Z}$.

We compare the exact result at time $n = 0$ with the approximations computed by the various algorithms for $d = 8$, $\{\lambda_1, \ldots, \lambda_8\} = \{\log 8, \log 7, \log 6, \ldots, \log 1\}$ and $\epsilon = 0.1$ for varying amounts of cocycle data $\{A(T^{-N}x), \ldots, A(T^N x)\}$. The exact model has a well separated spectrum, is generated using invertible matrices, and is of relatively low dimension.

For this model we use the following choice of parameters to execute the algorithms:

- Algorithm 2.2: $M = N$ and $\{N_k\} = \{1, 6, \ldots, 5k - 4, \ldots, 5K - 4, N\}$ where $5K - 4 < N \leq 5K + 1$.
- Algorithm 3.1: We estimate the three largest Lyapunov exponents $\lambda_1 > \lambda_2 > \lambda_3$ and set $\Lambda^{\text{right}} = \lambda_2 + 0.1(\lambda_1 - \lambda_2)$ and $\Lambda^{\text{left}} = \lambda_2 - 0.1(\lambda_2 - \lambda_3)$.
- Algorithm 3.2: As for Algorithm 3.1.
- Algorithm 4.5: $M = N$, $M' = 5$, and $c' = (0, 1)$.
- Algorithm 4.6: Let $M_1 = N$, $M_1' = 5$ and $M_2 = N$.

Fig. 4 compares the approximations yielded from the various approaches outlined in Sections 2–4 with the known solution of Eq. (31). Each algorithm exhibits approximately exponential convergence with respect to the length of the sample cocycle up to (almost) machine accuracy of about 10^{-16}. Algorithm 4.3 is notably erratic whereas the other algorithms converge smoothly, which suggests that in the limited data scenario (small N) it represents the less satisfactory choice.
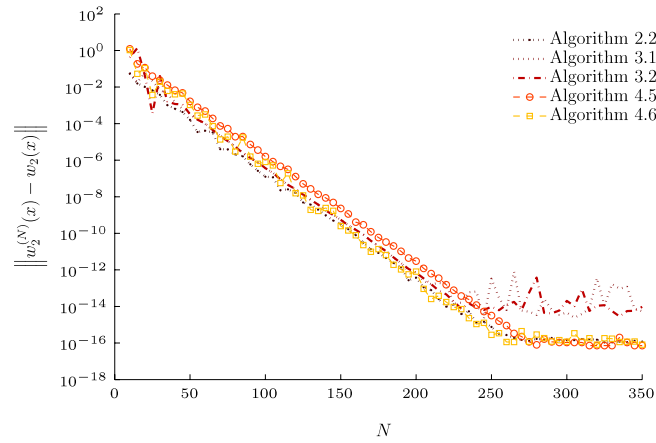
Fig. 4. Comparing the approximations of the second Oseledets subspace $W_2^{(N)}(x)$ with the exact solution $W_2(x)$ which is known *a priori*. Each "N-approximation" is computed using cocycle data $\{A(T^{-N}x), A(T^{-N+1}x), \ldots, A(T^Nx)\}$. The comparison is simply the Euclidean norm of the separation of the two unit vectors $w_2^{(N)}(x) \in W_2^{(N)}(x)$ and $w_2(x) \in W_2(x)$.
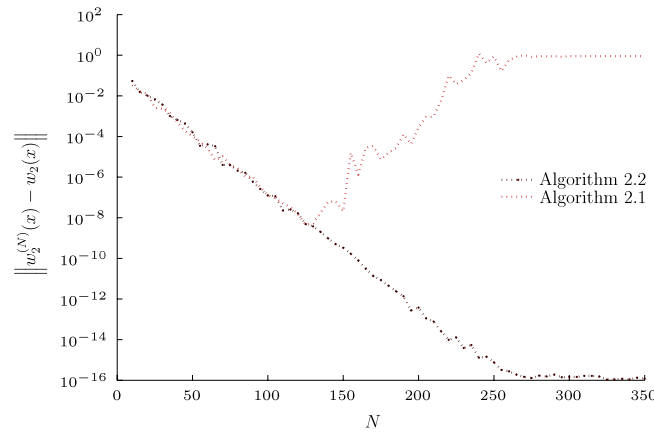

Fig. 5. Comparing the approximation of the second Oseledets vector with the exact solution for the two SVD based approaches, demonstrating the numerical instability which is overcome in the adapted SVD approach.

Performing a linear regression on the data shown in Fig. 4 for a sample where each method is still converging ($N \lesssim 230$) yields a gradient of $-0.13342 \approx \log 7 - \log 8$ suggesting that the convergence rate is roughly exponential in the difference of the largest and 2nd largest Lyapunov exponent.

Algorithm 2.2 is slightly more accurate than the other algorithms for large N, while there is a limit to the accuracy of Algorithms 3.1 and 3.2.

It is worth noting that Algorithms 2.2 and 4.6 do not perform as well when approximating Oseledets subspaces corresponding to Lyapunov exponents $\lambda_3, \ldots, \lambda_\ell$. Whilst they reach machine accuracy with ease for W_1 and W_2, forming $A(x, n) = A(T^{n-1}x) \cdots A(x)$ via numerical matrix multiplication produces greater inaccuracies for subspaces W_3 through W_ℓ (see Remark 1). On the other hand, Algorithms 3.1, 3.2 and 4.5 do not suffer from the same issue because they do not need to form $A(x, n)$ but use only the generating matrices $A(x)$. As such, they still reach machine accuracy, although a greater amount of data (larger N) is required.

Fig. 5 is similar to Fig. 4 except that it compares only Algorithms 2.1 and 2.2. In doing so, it highlights the result of one of the numerical instabilities of Algorithm 2.1, namely the pushing forward of $U_j^{(M)}(T^{-N}x)$ in Step 3.

Finally, Fig. 6 shows the execution times of Algorithms 2.2, 3.1, 3.2, 4.5 and 4.6, which were timed using MATLAB's timing functionality. The most time-consuming step in Algorithm 4.5 is the SVD performed as part of the alterations from Section 4.1.1. Algorithm 4.6 must perform two SVDs and Algorithm 2.2 must perform many more, which accounts for their longer execution times.

5.2. Case study 2: particle dynamics—two disks in a quasi-one-dimensional box

We consider the quasi-one-dimensional heat system studied extensively by Morriss et al. [4,3,20–23] which consists of two disks of diameter $\sigma = 1$ in a rectangular box, $[0, L_x] \times [0, L_y]$, in which the shorter side, has length $L_y < 2\sigma$ so that the disks may not change order. The two disks interact elastically with each other and the short walls, but periodic boundary conditions are enforced in the y-direction. The phase space of the system is then the set $X \subset \mathbb{R}^8$

$$X = \left(([0, L_x] \times [0, L_y]) / \sim\right)^2 \times \mathbb{R}^2 \times \mathbb{R}^2,$$
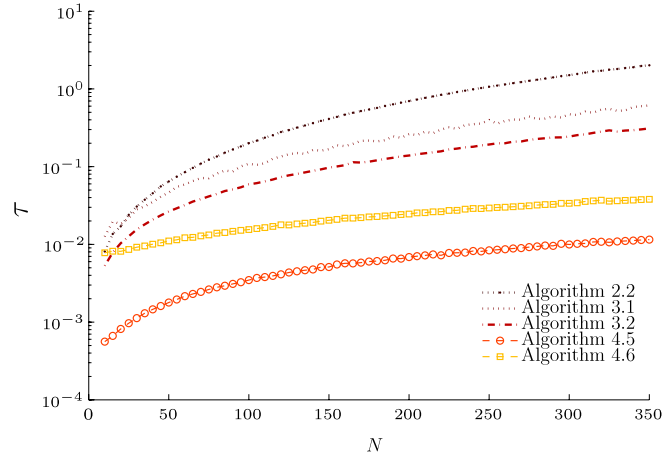
Fig. 6. Comparing the execution time τ of the various algorithms using MATLAB's timing functionality. Each algorithm is executed using the cocycle data $\{A(T^{-N}x), \ldots, A(T^N x)\}$.

where $\sim$ is the equivalence class associated with periodic boundary conditions, that is, $(x_1, y_1), (x_2, y_2) \in [0, L_x] \times [0, L_y]$ have $(x_1, y_1) \sim (x_2, y_2)$ if $y_1 = y_2 \mathrm{mod} L_y$ and $x_1 = x_2$.

The flow $\phi^\tau : X \to X$ consists of free-flight maps of time τ, $\mathcal{F}^\tau : X \to X$, and collision maps $\mathcal{C} : X \to X$ so that $\phi^\tau(x) = \mathcal{C} \circ \mathcal{F}^{\tau_n} \circ \cdots \circ \mathcal{F}^{\tau_2} \circ \mathcal{C} \circ \mathcal{F}^{\tau_1}(x)$ where $\tau_1 + \cdots + \tau_n = \tau$. We consider a discrete-time version of the system by mapping from the instant after collision to the instant after the next collision, that is $x \mapsto \mathcal{C} \circ \mathcal{F}^{\tau(x)}(x)$ where $\tau(x)$ is the free-flight time in the continuous system. The matrix cocycle is generated by the 8×8 Jacobian matrices or the derivative of the flow evaluated instantly after each collision (i.e. $A(x) = D\left(\mathcal{C} \circ \mathcal{F}^{\tau(x)}\right)(x)$, see [4] for details). Due to a number of dynamic symmetries the system has Lyapunov exponents $\lambda_1 > \lambda_2 > 0 > -\lambda_2 > -\lambda_1$ with multiplicities 1, 1, 4, 1 and 1 respectively. This system has some symmetry, a high variation in expansion rates from iteration to iteration, a well separated spectrum, invertible Jacobian matrices, and relatively low dimension. Numerical integration of an orbit consisting of 4646 collisions yielded a sequence of Jacobian matrices $\{A(T^{-2323}x), \ldots, A(T^{2322}x)\}$ which generate the cocycle A.

For this model we use the same choice of parameters to execute the algorithms as with the previous model:

- Algorithm 2.2: $M = N$ and $\{N_k\} = \{1, 6, \ldots, 5k - 4, \ldots, 5K - 4, N\}$ where $5K - 4 < N \le 5K + 1$.
- Algorithm 3.1: We estimate the three largest Lyapunov exponents $\lambda_1 > \lambda_2 > \lambda_3$ and set $\Lambda^{\mathrm{right}} = \lambda_2 + 0.1(\lambda_1 - \lambda_2)$ and $\Lambda^{\mathrm{left}} = \lambda_2 - 0.1(\lambda_2 - \lambda_3)$.
- Algorithm 3.2: As for Algorithm 3.1.
- Algorithm 4.5: $M = N$, $M' = 5$, and $c' = (0, 1)$.
- Algorithm 4.6: Let $M_1 = N$, $M'_1 = 5$ and $M_2 = N$.

5.2.1. Criteria to assess the accuracy of estimated Oseledets spaces

Since the Oseledets subspaces for this model are unknown, we test the approximations for two properties of Oseledets subspaces, namely their equivariance and the expansion rate, which defines the corresponding Lyapunov exponent.

Equivariance: To test for equivariance, we approximate the second Oseledets vector, $w_2^{(N)}(T^n x)$, at each time $n = 0, 1, \ldots, 30$. We then compute $\left\| \mathcal{N}\left(A(x, n)w_2^{(N)}(x)\right) - w_2^{(N)}(T^n x)\right\|$ and plot the result, where $v \overset{\mathcal{N}}{\mapsto} v/\|v\|$. If the approximations are equivariant this value would be zero.

Expansion rate: To test the expansion rate, each approach is used to compute the second Oseledets vector, $w_2^{(N)}(x) \in W_2^{(N)}(x)$, at time $n = 0$ and we plot $\frac{1}{m} \log \left\| A(x, m)w_2^{(N)}(x)\right\|$ versus m. If $W_2^{(N)}(x)$ is accurate, elements of $W_2^{(N)}(x)$ should grow at the correct rate: λ_2.

Whilst the Oseledets vector $w_2(x)$ must satisfy the above two properties, we must be careful when examining the results of these numerical experiments. For instance, (i) it is possible to choose vectors that are equivariant despite not being contained in any single Oseledets subspace, and (ii) any element of $V_2(x) \setminus V_3(x) \supsetneq W_2(x)$ (a much larger set than $W_2(x)$) has Lyapunov exponent λ_2.

5.2.2. Numerical results

Fig. 7 shows the results of the equivariance test for the quasi-one-dimensional two disk model. At the lower end of cocycle data length ($N = 75$) all Algorithms except 3.2 display reasonable equivariance, although Algorithm 3.1 remains equivariant for only a handful of steps. For $N = 150$ and $N = 225$ all approaches appear to produce close to equivariant results (note the changing scales in the vertical direction), with Algorithms 3.1 and 3.2 lagging behind when $N = 225$.

Fig. 8 shows the results of the expansion rate test for the quasi-one-dimensional two disk model for various amounts of cocycle data $\{A(T^{-N}x), \ldots, A(T^N x)\}$. As expected, when there is a limited amount of data available (N small) the approximations either expand at the higher rate of λ_1 or only expand at the rate of λ_2 for a brief time before the error grows too large. As N is increased, the approximations expand at λ_2 for longer periods, suggesting that they more accurately represent $w_2(x)$.
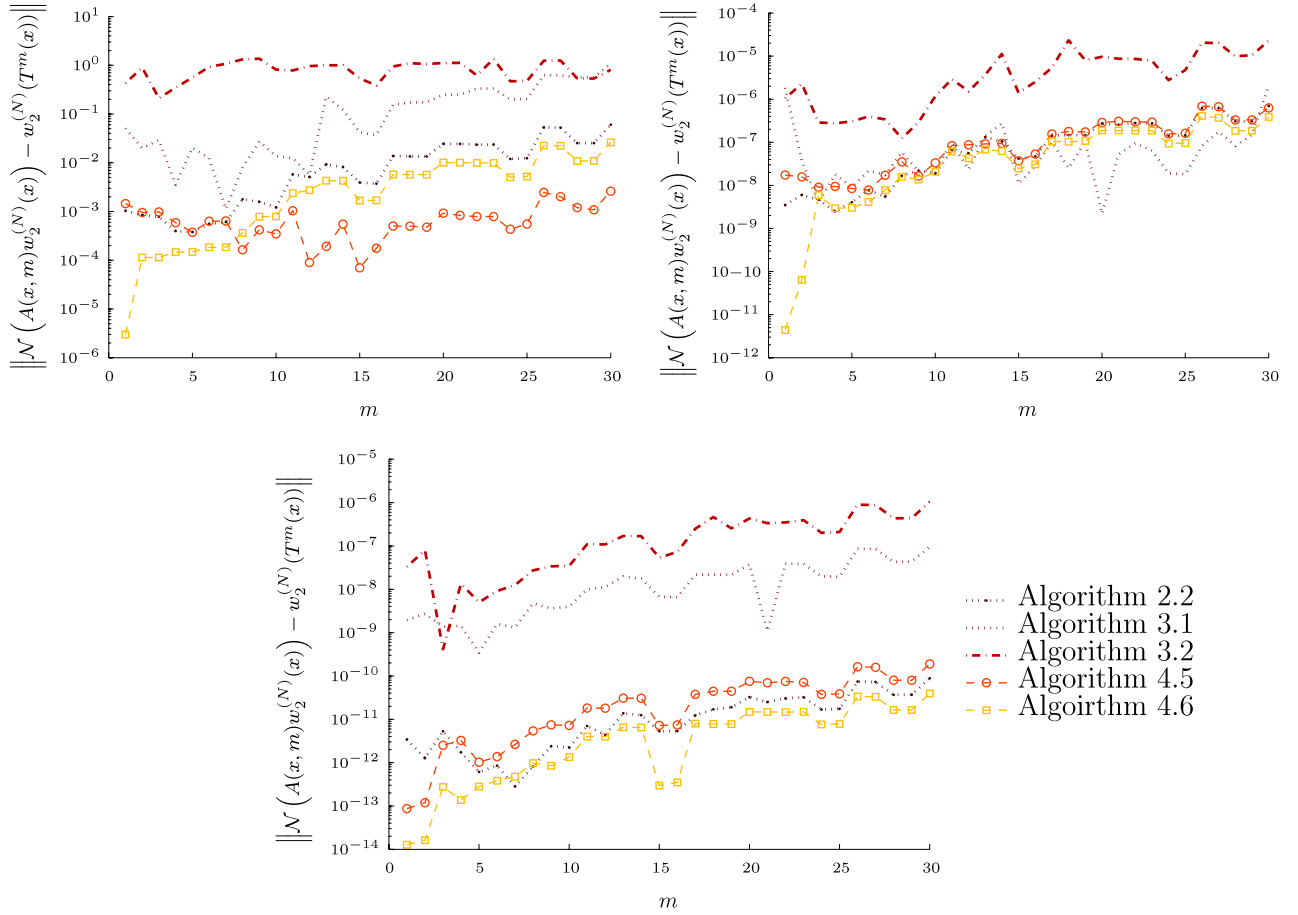
Fig. 7. The equivariance test for the various algorithms on the quasi-one-dimensional two disk model. Each approach is used to approximate the second Oseledets vector, $w_2^{(N)}(T^n x) \in W_2^{(N)}(T^n x)$ using cocycle data $\{A(T^{-N}x), \ldots, A(T^N x)\}$, at each time $n = 0, 1, \ldots, 30$. We then compute $\left\| \mathcal{N} A(x, n) w_2^{(N)}(x) - w_2^{(N)}(T^n x) \right\|$ and plot the result. Note the different scales on each vertical axis. The plots shown are for $N = 75$ (top left), $N = 150$ (top right) and $N = 225$ (bottom).

Most algorithms perform similarly regarding expansion rate. Note that the amount of cocycle data (size of N) needed to perform well in the expansion rate test is less than that needed to perform well in the Equivariance test—this demonstrates the importance of good performance in both tests in order to assess whether or not the algorithms are performing well.

5.3. Case study 3: time-dependent fluid flow in a cylinder; a transfer operator description

An important emerging application for Oseledets subspaces is the detection of strange eigenmodes, persistent patterns, and coherent sets for aperiodic time-dependent fluid flows. In the periodic setting strange eigenmodes have been found as eigenfunctions of a Perron–Frobenius operator via classical Floquet theory; [41–43]. However, in the aperiodic time-dependent setting, Floquet theory cannot be applied. An extension to aperiodically driven flows was derived in [6], based on the new multiplicative ergodic theory of [2]. Discrete approximations of a Perron–Frobenius cocycle representing the aperiodic flow are constructed and in this aperiodic setting the leading sub-dominant Oseledets subspaces play the role of the leading sub-dominant eigenfunctions in the periodic forcing case.

We review the four methods of approximating Oseledets subspaces with the aperiodically driven cylinder flow from [6]. The flow domain is $Y = [0, 2\pi] \times [0, \pi]$, $t \in \mathbb{R}^+$ and the flow is defined by the following forced ODE:

$$
\begin{aligned}
\dot{x} &= c - \tilde{A}(\tilde{z}(t)) \sin(x - v\tilde{z}(t)) \cos(y) + \varepsilon G(g(x, y, \tilde{z}(t))) \sin(\tilde{z}(t)/2) \quad \mathrm{mod}\, 2\pi, \\
\dot{y} &= \tilde{A}(\tilde{z}(t)) \cos(x - v\tilde{z}(t)) \sin(y).
\end{aligned}
\tag{32}
$$

Here, $\tilde{z}(t) = 6.6685 z_1(t)$, where $z_1(t)$ is generated by the standard Lorenz flow, $\tilde{A}(\tilde{z}(t)) = 1 + 0.125 \sin(\sqrt{5}\tilde{z}(t))$, $G(\psi) := 1/(\psi^2 + 1)^2$ and the parameter function $\psi = g(x, y, \tilde{z}(t)) := \sin(x - v\tilde{z}(t)) \sin(y) + y/2 - \pi/4$ vanishes at the level set of the stream function of the unperturbed ($\varepsilon = 0$) flow at instantaneous time $t = 0$, i.e., $s(x, y, 0) = \pi/4$, which divides the phase space in half.

We set $\varepsilon = 1$ as this value is sufficiently large to ensure no KAM tori remain in the jet regime, but sufficiently small to maintain islands originating from the nested periodic orbits around the elliptic points of the unperturbed system.

We construct the discretized Perron–Frobenius matrices $P_x^{(\tau)}(t) =: A(x)$ as described in Section 3 of [6], and briefly recapped in Example 1.2, using a uniform grid of 120×60 boxes, $\tau = 8$ and $-32 \leq t \leq 32$. In total, we generate 8 such matrices of dimension 7200×7200. Thus, in this case study we have a limited amount of data, no symmetry, high dimension, and the matrices are non-invertible and sparse.
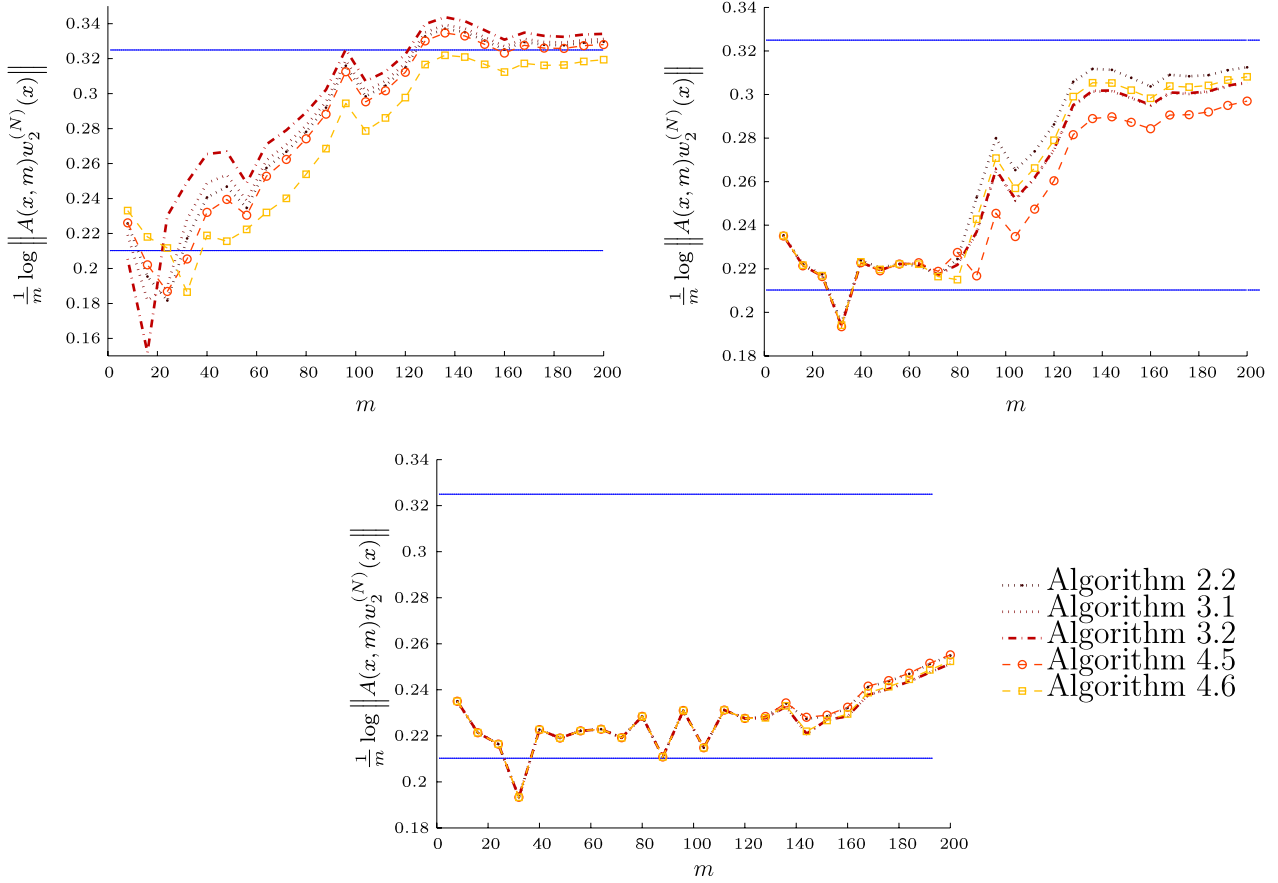
Fig. 8. The expansion rate test for the various approaches on the quasi-one-dimensional two disk model. The second Oseledets vector, $w_2^{(N)}(x) \in W_2^{(N)}(x)$, is approximated using cocycle data $\{A(T^{-N}x), \ldots, A(T^N x)\}$ and we plot $\frac{1}{m} \log \left\| A(x, m) w_2^{(N)}(x) \right\|$ versus m. If the approximation is accurate this quantity should tend to the value of $\lambda_2 \approx 0.210$, otherwise it would tend to the value of $\lambda_1 \approx 0.325$ both of which are shown in blue. The plots shown are for $N = 25$ (top left), $N = 75$ (top right) and $N = 150$ (bottom).

In order to obtain reasonable results we executed Algorithms 2.2, 3.1, 3.2, 4.3 and 4.6 with the following parameters:

- Algorithm 2.2: $M = N = 4$ and $\{N_k\} = \{2, 4\}$.
- Algorithm 3.1: We estimate the three largest Lyapunov exponents $\lambda_1 > \lambda_2 > \lambda_3$ and set $\Lambda^{\text{right}} = \lambda_2 + 0.1(\lambda_1 - \lambda_2)$ and $\Lambda^{\text{left}} = \lambda_2 - 0.1(\lambda_2 - \lambda_3)$.
- Algorithm 3.2: As for Algorithm 3.1.
- Algorithm 4.5: $M' = M = 4$ (so that only an SVD is used, and no push-forward step), $N = 4$ and $c' = (0, 1)$.
- Algorithm 4.6: $M_1 = M_1' = 4$ and $M_2 = 4$.

The results of these numerical experiments are shown in Figs. 9 and 10. Recall that in this setting, the cocycle $A(x, n)$ is a cocycle of discretized Perron–Frobenius operators acting on piecewise constant functions defined on Y; we identify these piecewise constant functions (with 7200 pieces) with vectors in $\mathbb{R}^{7200}$. Fig. 9 first shows the approximations of the second Oseledets vector $w_2(x)$ at time $t = 0$. In this setting the Oseledets vectors locate *coherent structures*: Fig. 10 compares the push-forward of the approximations in Fig. 9 with independently computed approximations of $w_2(Tx)$—the second Oseledets vector at time $t = 8$.

In this study the data sample is insufficiently long for Algorithm 3.2 to work effectively, but the other algorithms produce similar results. A visual inspection of Fig. 10 shows that the highlighted structures are approximately equivariant/coherent.

6. Conclusion

We introduced two new methods for computing Oseledets subspaces: one based on singular value decompositions and the other based on dichotomy projectors. We also reviewed recent methods by Ginelli et al. [36] and Wolfe and Samelson [37], and presented an improvement to both of these approaches that intelligently selected initial bases when only short time series were available to compute with. Finally, we carried out a comparative numerical investigation involving all four methods.

Generally speaking, we found that Algorithms 2.2, 4.5 and 4.6 outperformed the dichotomy projector methods (Algorithms 3.1 and 3.2) when limited to moderate amounts of data were available, however, the dichotomy projector methods performed very well when long time series of matrices were available. The Ginelli approach (in particular the improved Algorithm 4.5) also worked very well with long time series.

The improvements made to Algorithm 2.1 in Section 2.1 (namely the orthogonalization step in Algorithm 2.2) produced an algorithm that could take advantage of longer matrix sequences and return very accurate results. Of course, for each Algorithm one must choose the associated parameters sensibly to ensure good results.
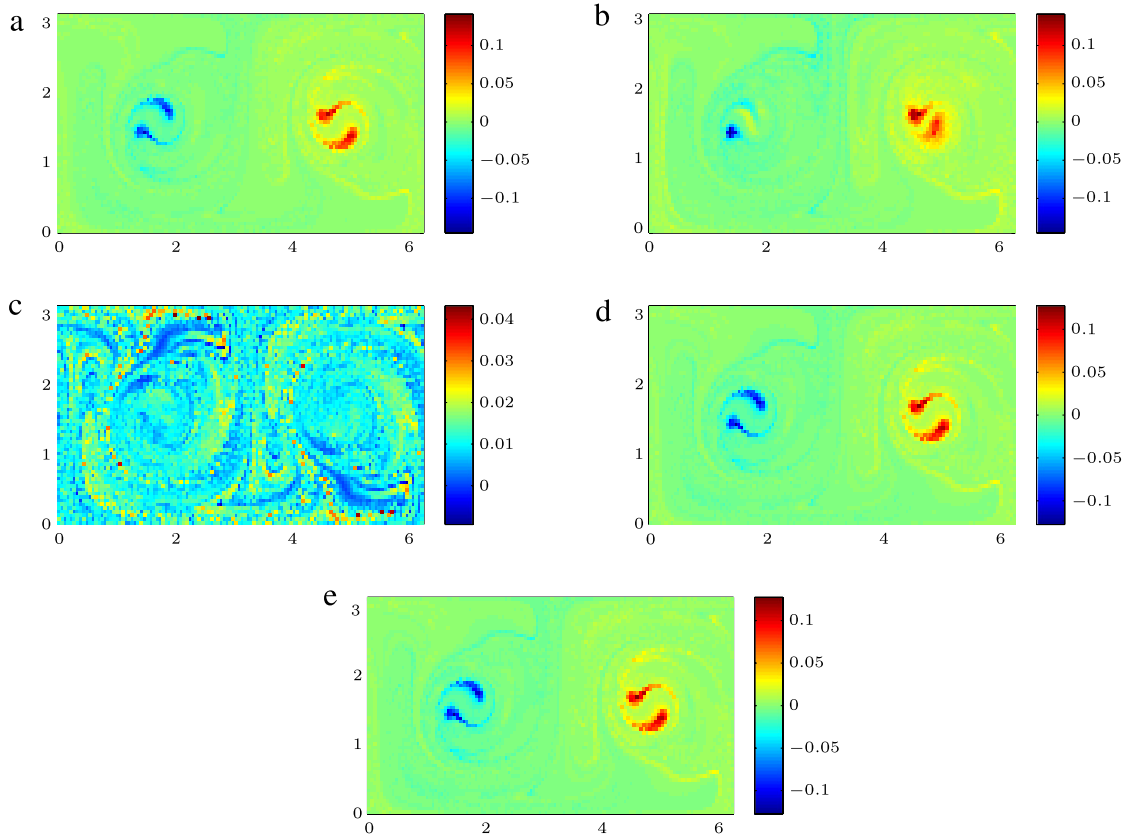
Fig. 9. The second Oseledets subspace as determined by (a) Algorithm 2.2, (b) Algorithm 3.1, (c) Algorithm 3.2, (d) Algorithm 4.3 and (e) Algorithm 4.6.

When only a short to moderate time series was available, we found mixed results in terms of the best algorithm. The improved SVD approach (Algorithm 2.2) was best for low to moderate length time series in the exact Toy model, while the improved Ginelli (Algorithm 4.5) and improved Wolfe (Algorithm 4.6) were marginally best in terms of equivariance and expansion rate, respectively for the 2-disk model. Each of these three algorithms produced similar results in the fluid-flow system.

Choosing appropriate parameters for a particular application can be difficult. In the present review, good values were chosen by educated experimentation. On the other hand, the dichotomy projector methods, Algorithms 3.1 and 3.2, use parameters (Λ^{right} and Λ^{left}) which can be chosen in a deterministic manner—by estimating Lyapunov exponents, which is a reasonably robust numerical procedure. Furthermore, a rigorous error approximation exists for Algorithm 3.2, a feature currently lacking for Algorithms 2.2, 4.5 and 4.6.

The memory footprint of each approach scales quite differently with dimension. In Section 5.3, Algorithms 2.2 and 4.6 could take advantage of the sparseness of the $d \times d$ generating matrices of the cocycle. However, since $A(x, n)$ is formed by matrix multiplication, for large n the matrix $A(x, n)$ becomes dense and may require memory of the order of d^2 floating point numbers. The dichotomy projector Algorithms 3.1 and 3.2, need to form an $Nd \times (N + 1)d$ matrix, but with sparse generating matrices, this requires memory much less than of the order of d^2 floating point numbers. Algorithm 4.5 has the most conservative memory footprint, but depends on its initialization parameter M' and the Oseledets subspace number j. If M' is large, then $A(T^{-M}x, M')$ in Step 1 can become dense and require $\mathcal{O}(d^2)$ floating point numbers. On the other hand, the stationary Lyapunov basis requires jd floating point numbers to be stored, so if $j \approx d$ this can becomes comparable to d^2.

Section 5.3 involves non-invertible generating matrices and apart from Algorithm 3.2, each approach succeeded in producing a reasonable solution, showing that the Algorithms can perform well in the non-invertible setting. Continuing with the non-invertible situation, if one wishes to approximate Oseledets subspaces corresponding to negative numbers with very large magnitudes ($\lambda_j \approx -\infty$), then Algorithms 2.2 and 4.6 may struggle as rapidly contracting directions (relative to the dominant direction corresponding to λ_1) are quickly squashed during the matrix multiplication used to approximate $A(x, n)$ leading to inaccurate numerical representation of $A(x, n)$.

The dichotomy projector approaches of Algorithms 3.1 and 3.2 are able to compute Oseledets subspaces corresponding to smaller, sub-dominant Lyapunov exponents $\lambda_3, \lambda_4, \ldots$ provided larger amounts of cocycle data is available. However, if $\lambda_j \approx -\infty$, we are forced to choose Λ^{right} or $\Lambda^{\text{left}} \approx -\infty$ which means either problem (14) or (15) (in Algorithm 3.1 which also feature in Algorithm 3.2) are ill-conditioned and fail.

The same problem manifests itself in Algorithm 4.5, even though it is able to compute Oseledets subspaces corresponding to smaller, sub-dominant Lyapunov exponents. The sum of the logarithm of the diagonal entries of the $j \times j$ generating matrices of the cocycle $R(x, n)$ average to the logarithmic expansion rate of the j-parallelepiped formed at x by the stationary Lyapunov basis $s_1^{(\infty)}(x), \ldots, s_j^{(\infty)}(x)$ as it is pushed-forward. Thus, the logarithm of the ith diagonal entry of the generating matrices of $R(x, n)$ has a time average of λ_i [18] and if $\lambda_j \approx -\infty$, $R(x, n)$ will feature diagonal entries close to, or equal to zero and $R(x, n)^{-1}$ will not exist.

In summary, Algorithms 2.2 and 4.6 are best suited to situations with limited cocycle data when one of the most dominant Oseledets subspaces is desired. Algorithm 4.5 can be applied to both limited and high data situations by choosing M' appropriately, and can compute most Oseledets subspaces provided their Lyapunov exponents are well-conditioned. If ample data is available and information regarding
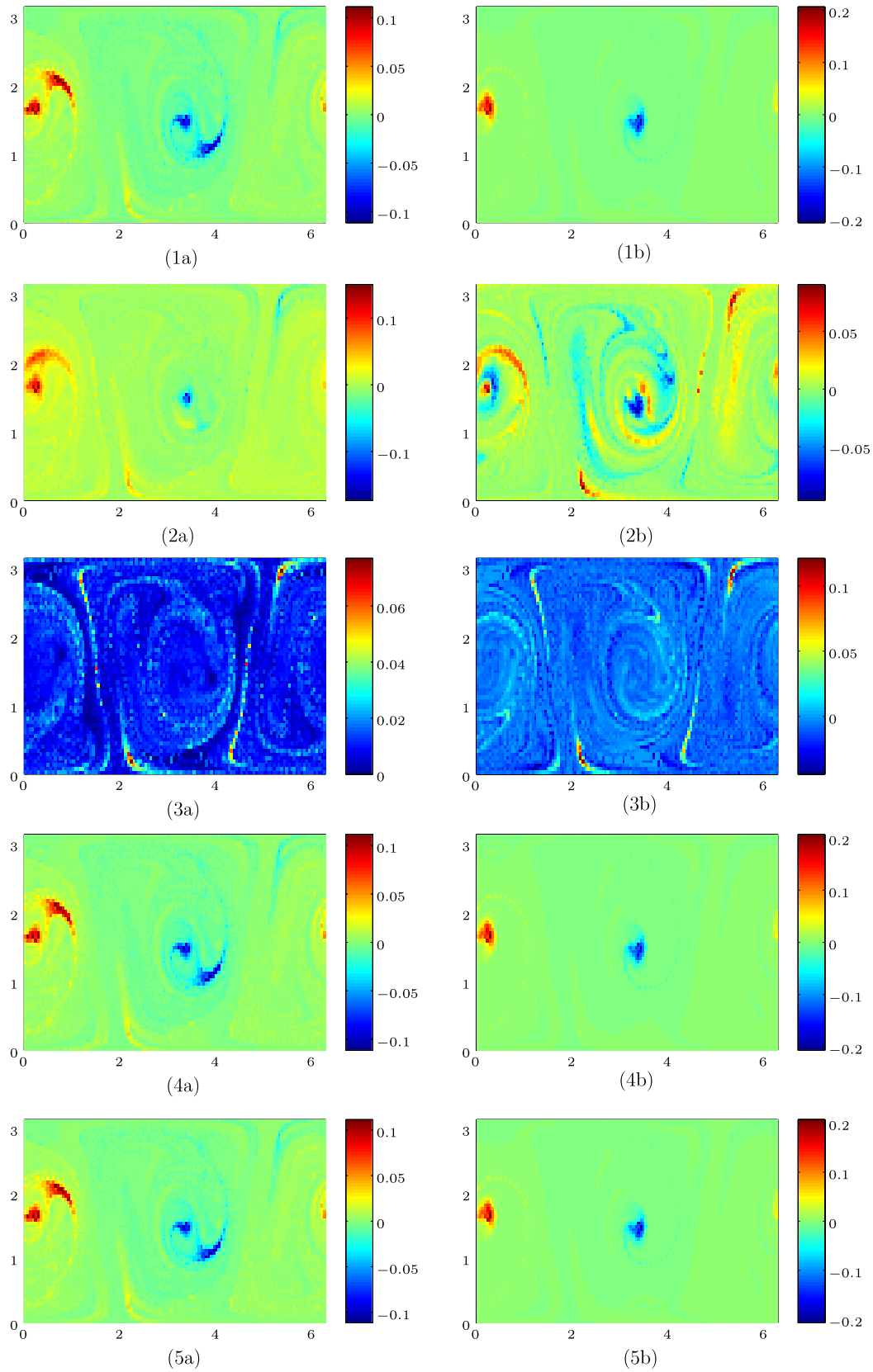
Fig. 10. Comparing the approximations of the second Oseledets vector $w_2(Tx)$ at time $t = 8$ with the push-forward of the approximations at time $t = 0$. Those labeled (a) are the push-forwards $A(x, 1)w_2^{(4)}(x)$ whilst those labeled (b) are independently computed approximations $w_2^{(4)}(Tx)$ of $w_2(Tx)$. The algorithms used are as follows: (1) Algorithm 2.2, (2) Algorithm 3.1, (3) Algorithm 3.2, (4) Algorithm 4.3 and (5) Algorithm 4.6.

the system is lacking (making the choice of parameters for the other approaches difficult), the approaches of Algorithms 3.1 and 3.2 may be preferred for their relatively deterministic parameter selection.

Acknowledgments

References

[1] F. Colonius, W. Kliemann, The dynamics of control, in: Systems & Control: Foundations & Applications, Birkhäuser Boston Inc., Boston, MA, 2000. With an appendix by Lars Grüne.
[2] G. Froyland, S. Lloyd, A. Quas, Coherent structures and isolated spectrum for Perron–Frobenius cocycles, Ergodic Theory Dynam. Systems 30 (03) (2010) 729–756.
[3] G.P. Morriss, D. Truant, Characterization of the Lyapunov modes for an equilibrium quasi-one-dimensional system, J. Stat. Mech. Theory Exp. 2009 (2) (2009) P02029.
[4] T. Chung, D. Truant, G.P. Morriss, From Lyapunov modes to their exponents for hard disk systems, Phys. Rev. E 81 (6) (2010) 066208.
[5] P. Cvitanović, R.L. Davidchack, E. Siminos, On the state space geometry of the Kuramoto–Sivashinsky flow in a periodic domain, SIAM J. Appl. Dyn. Syst. 9 (2010) 1–33.
[6] G. Froyland, S. Lloyd, N. Santitissadeekorn, Coherent sets for nonautonomous dynamical systems, Physica D 239 (16) (2010) 1527–1541.
[7] M. Dellnitz, O. Junge, On the approximation of complicated dynamical behaviour, SIAM J. Numer. Anal. 36 (2) (1999) 491–515.
[8] G. Froyland, M. Dellnitz, Detecting and locating near-optimal almost-invariant sets and cycles, SIAM J. Sci. Comput. 24 (6) (2003) 1839–1863.
[9] G. Froyland, Statistically optimal almost-invariant sets, Physica D 200 (2005) 205–219.
[10] G. Froyland, N. Santitissadeekorn, A. Monahan, Transport in time-dependent dynamical systems: finite-time coherent sets, Chaos 20 (2010) 043116.
[11] V.I. Oseledec, A multiplicative ergodic theorem. Lyapunov characteristic numbers for dynamical systems, Trans. Moscow Math. Soc. (1968).
[12] G. Froyland, S. Lloyd, A. Quas, A semi-invertible Oseledets theorem with applications to transfer operator cocycles, Discrete Contin. Dyn. Syst. (in press) http://arxiv.org/abs/1001.5313.
[13] L. Arnold, Random Dynamical Systems, Springer Verlag, 1998.
[14] L. Dieci, C. Elia, The singular value decomposition to approximate spectra of dynamical systems. Theoretical aspects, J. Differential Equations 230 (2) (2006) 502–531.
[15] L. Dieci, C. Elia, E. Van Vleck, Exponential dichotomy on the real line: SVD and QR methods, J. Differential Equations 248 (2) (2010) 287–308.
[16] L. Dieci, C. Elia, E. Van Vleck, Detecting exponential dichotomy on the real line: SVD and QR algorithms, BIT 51 (3) (2011) 555–579.
[17] G. Benettin, L. Galgani, A. Giorgilli, J.M. Strelcyn, All Lyapunov characteristic numbers are effectively computable, C. R. Acad. Sci. Paris 286 (1978) 431–433.
[18] G. Benettin, L. Galgani, A. Giorgilli, J.M. Strelcyn, Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems: a method for computing all of them. Part I: theory. Part II: numerical application, Meccanica 15 (1980) 9–30.
[19] I. Shimada, T. Nagashima, A numerical approach to ergodic problem of dissipative dynamical systems, Progr. Theoret. Phys. 61 (1979) 1605–1616.
[20] D.J. Robinson, G.P. Morriss, Lyapunov mode dynamics in hard-disk systems, J. Stat. Phys. 131 (1) (2008) 1–31.
[21] T. Taniguchi, G.P. Morriss, Lyapunov modes for a nonequilibrium system with a heat flux, C.R. Phys. 8 (5–6) (2007) 625–632.
[22] T. Taniguchi, G.P. Morriss, Time-dependent mode structure for Lyapunov vectors as a collective movement in quasi-one-dimensional systems, Phys. Rev. E 71 (1) (2005) 016218.
[23] T. Taniguchi, G.P. Morriss, Time-oscillating Lyapunov modes and the momentum autocorrelation function, Phys. Rev. Lett. 94 (15) (2005) 154101.
[24] S.V. Ershov, A.B. Potapov, On the concept of stationary Lyapunov basis, Physica D 118 (3–4) (1998) 167–198.
[25] R.J. Sacker, G.R. Sell, A spectral theory for linear differential systems, J. Differential Equations 27 (3) (1978) 320–358.
[26] W.A. Coppel, Dichotomies in Stability Theory, in: Lecture Notes in Mathematics, vol. 629, Springer-Verlag, Berlin, 1978.
[27] K.J. Palmer, Exponential dichotomies, the shadowing lemma and transversal homoclinic points, in: Dynamics Reported, Vol. 1, Teubner, Stuttgart, 1988, pp. 265–306.
[28] B. Aulbach, S. Siegmund, The dichotomy spectrum for noninvertible systems of linear difference equations, J. Difference Equ. Appl. 7 (6) (2001) 895–913. On the occasion of the 60th birthday of Calvin Ahlbrandt.
[29] B. Aulbach, S. Siegmund, A spectral theory for nonautonomous difference equations, in: New Trends in Difference Equations (Temuco, 2000), Taylor & Francis, London, 2002, pp. 45–55.
[30] L. Dieci, E.S. Van Vleck, Lyapunov spectral intervals: theory and computation, SIAM J. Numer. Anal. 40 (2) (2002) 516–542 (electronic).
[31] L.Ya. Adrianova, Introduction to Linear Systems of Differential Equations, in: Translations of Mathematical Monographs, vol. 146, American Mathematical Society, Providence, RI, 1995. Translated from the Russian by Peter Zhevandrov.
[32] K.J. Palmer, Exponential dichotomy, integral separation and diagonalizability of linear systems of ordinary differential equations, J. Differential Equations 43 (2) (1982) 184–203.
[33] K.J. Palmer, Exponential separation, exponential dichotomy and spectral theory for linear systems of ordinary differential equations, J. Differential Equations 46 (3) (1982) 324–345.
[34] N.D. Cong, D.T. Son, An open set of unbounded cocycles with simple Lyapunov spectrum and no exponential separation, Stoch. Dyn. 7 (3) (2007) 335–355.
[35] T. Hüls, Numerical computation of dichotomy rates and projectors in discrete time, Discrete Contin. Dyn. Syst. Ser. B 12 (1) (2009) 109–131.
[36] F. Ginelli, P. Poggi, A. Turchi, H. Chaté, R. Livi, A. Politi, Characterizing dynamics with covariant Lyapunov vectors, Phys. Rev. Lett. 99 (13) (2007).
[37] C.L. Wolfe, R.M. Samelson, An efficient method for recovering Lyapunov vectors from singular vectors, Tellus Ser. A 59 (3) (2007) 355–366.
[38] T. Hüls, Computing Sacker–Sell spectra in discrete time dynamical systems, SIAM J. Numer. Anal. 48 (6) (2010) 2043–2064.
[39] N. Shinozaki, M. Sibuya, K. Tanabe, Numerical algorithms for the Moore–Penrose inverse of a matrix: direct methods, Ann. Inst. Statist. Math. 24 (1972) 193–203.
[40] P.V. Kuptsov, U. Parlitz, Theory and computation of covariant Lyapunov vectors, J. Nonlin. Sci. 22 (5) (2012) 727–762.
[41] A. Pikovsky, O. Popovych, Persistent patterns in deterministic mixing flows, Europhys. Lett. 61 (5) (2003) 625.
[42] W. Liu, G. Haller, Strange eigenmodes and decay of variance in the mixing of diffusive tracers, Physica D 188 (1) (2004) 1–39.
[43] O. Popovych, A. Pikovsky, B. Eckhardt, Abnormal mixing of passive scalars in chaotic flows, Phys. Rev. E 75 (3) (2007) 036308.