Stable Rank-Adaptive Dynamically Orthogonal Runge–Kutta Schemes

Citation: Charous, Aaron and Lermusiaux, Pierre F. J. 2024. "Stable Rank-Adaptive Dynamically Orthogonal Runge–Kutta Schemes." SIAM Journal on Scientific Computing, 46 (1).

As Published: 10.1137/22m1534948

Publisher: Society for Industrial & Applied Mathematics (SIAM)

Persistent URL: https://hdl.handle.net/1721.1/153761

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

STABLE RANK-ADAPTIVE DYNAMICALLY ORTHOGONAL RUNGE-KUTTA SCHEMES[*]

AARON CHAROUS[†] AND PIERRE F.J. LERMUSIAUX[†]

Abstract. We develop two new sets of stable, rank-adaptive Dynamically Orthogonal Runge-Kutta (DORK) schemes that capture the high-order curvature of the nonlinear low-rank manifold. The DORK schemes asymptotically approximate the truncated singular value decomposition at a greatly reduced cost while preserving mode continuity using newly derived retractions. We show that arbitrarily high-order optimal perturbative retractions can be obtained, and we prove that these new retractions are stable. In addition, we demonstrate that repeatedly applying retractions yields a gradient-descent algorithm on the low-rank manifold that converges superlinearly when approximating a low-rank matrix. When approximating a higher-rank matrix, iterations converge linearly to the best low-rank approximation. We then develop a rank-adaptive retraction that is robust to overapproximation. Building off of these retractions, we derive two rank-adaptive integration schemes that dynamically update the subspace upon which the system dynamics are projected within each time step: the stable, optimal Dynamically Orthogonal Runge-Kutta (so-DORK) and gradient-descent Dynamically Orthogonal Runge-Kutta (gd-DORK) schemes. These integration schemes are numerically evaluated and compared on an ill-conditioned matrix differential equation, an advection-diffusion partial differential equation, and a nonlinear, stochastic reaction-diffusion partial differential equation. Results show a reduced error accumulation rate with the new stable, optimal and gradient-descent integrators. In addition, we find that rank adaptation allows for highly accurate solutions while preserving computational efficiency.

Key words. Dynamical low-rank approximation, DO equations, matrix differential equations, stochastic differential equations, uncertainty quantification, reduced-order modeling, differential geometry, retraction, fixed-rank matrix manifold

AMS subject classifications. 54C15, 65F55, 53B21, 15A23, 57Z05, 57Z20, 57Z25, 60G60, 65C30, 65M12, 65M22, 65C20, 81S22, 94A08, 53A07, 35R60

1. Introduction.

Scientific computing often involves numerical simulations which overwhelm available computational resources. A common reduced-order modeling technique is to use low-rank approximations of matrices and/or tensors to improve computational efficiency and reduce storage requirements. Mathematically, we seek $X(t)$, a low-rank approximation to our full state $\mathfrak{X}(t)$, such that at all discrete times t_i, $X(t_i)$ is the best approximation to $\mathfrak{X}(t_i)$. That is,

$$X(t_i) = \underset{\tilde{X}(t_i)\in\mathscr{M}_r}{\arg\min} \, ||\tilde{X}(t_i) - \mathfrak{X}(t_i)|| \quad \forall t_i,$$

where $\mathscr{M}_r$ denotes the manifold of rank-r matrices (see Table 1 for key notation).

The dynamical low-rank approximation (DLRA) [33] is a method to instantaneously optimally evolve a system's low-rank approximation for common time-dependent partial differential equations (PDEs) [17]. Suppose we are given a discretized dynamical system, stochastic or deterministic, as a first-order differential equation

$$\frac{d\mathfrak{X}}{dt} = \mathscr{L}(\mathfrak{X}, t; \omega), \tag{1.1}$$

where $\omega \in \Omega$ denotes a simple event in a stochastic event space. Note that any order differential equation may be rewritten as a first-order system, so this is not at all restrictive (see, e.g., [30]). There are two common approaches to solve for X. The first is to derive an instantaneously optimal differential equation for X in terms of $\mathscr{L}$ given that X is restricted to the low-rank manifold. Using the Dirac-Frenkel time-dependent variational principle (see Table

[*] We thank the Office of Naval Research for partial support under grants N00014-19-1-2693 (IN-BDA) and N00014-19-1-2664 (Task Force Ocean, DEEP-AI).

[†] Mechanical Engineering, Center for Computational Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 (acharous@mit.edu, pierrel@mit.edu)

2 for the incurred approximation error), we arrive at the following differential equation for our low-rank approximation.

$$\frac{dX}{dt} = \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r}[\mathscr{L}(X,t;\omega)] \tag{1.2}$$

For a derivation, we refer to [33, 17, 9, 12]. Next, a parameterization of the low-rank matrix X is chosen; two common parameterizations are $X = UZ^T$ with orthonormal modes $U \in \mathcal{V}_{m,r}$ and coefficients $Z \in \mathbb{R}_*^{n\times r}$ as in [17, 16, 11, 12], or $X = USV^T$ similar to a classic singular value decomposition (SVD) with $U \in \mathcal{V}_{m,r}$, $S \in \mathbb{R}_*^{r\times r}$, and $V \in \mathcal{V}_{n,r}$ as in [33, 49, 43, 38, 39]. With the parameterization, evolution equations for U and Z (or U, S, and V) are derived with a choice of gauge in order to eliminate redundant degrees of freedom and simplify the equations. For instance, the dynamically orthogonal (DO) condition insists that the rate-of-change $\dot{U} \in \mathcal{U}_{m,r}$ (see table 1) which, from (1.2), yields the following equations.

$$\begin{aligned} \dot{U} &= \mathscr{P}_U^{\perp}\mathscr{L}(UZ^T,t;\omega)Z(Z^TZ)^{-1} \\ \dot{Z} &= \mathscr{L}(UZ^T,t;\omega)^T U \end{aligned} \tag{1.3}$$

Alternatively, insisting that $\dot{U} \in \mathcal{T}_U\mathcal{V}_{m,r}$ and $\dot{V} \in \mathcal{T}_V\mathcal{V}_{n,r}$ yields the following differential equations for U, S, and V.

$$\begin{aligned} \dot{U} &= \mathscr{P}_U^{\perp}\mathscr{L}(USV^T,t;\omega)VS^{-1} \\ \dot{S} &= U^T\mathscr{L}(USV^T,t;\omega)V \\ \dot{V} &= \mathscr{P}_V^{\perp}\mathscr{L}(USV^T,t;\omega)^T US^{-T} \end{aligned} \tag{1.4}$$

Finally, different numerical schemes may be employed to solve (1.3) or (1.4). Numerical integration schemes include the projector-splitting integrator [49], its closely related variant [8], and similar approaches based in local coordinates [4, 5].

A second approach is to employ so-called projection methods (see, e.g., [25, 26]). Instead of numerically integrating $\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\mathscr{L}$, the full $\mathscr{L}$ is integrated and then projected back onto the low-rank. Integrating (1.1) in time, we obtain

$$\mathfrak{X}_{i+1} = \mathfrak{X}_i + \Delta t\overline{\mathscr{L}}. \tag{1.5}$$

Here, $\overline{\mathscr{L}} = \frac{1}{\Delta t}\int_{t_i}^{t_{i+1}} \mathscr{L}(\mathscr{X}(t),t;\omega)dt + \mathcal{O}(\Delta t^k)$ is either the exactly integrated differential operator or the approximately integrated differential operator up to order k. In the exact case, the integrand $\mathscr{L}$ in $\overline{\mathscr{L}}$ depends on the unknown future solution $\mathscr{X}$; furthermore, $\mathscr{X}$ may be the full-rank system $\mathfrak{X}$ or the low-rank system X, though the latter is another approximation inducing a dynamical model closure error [12], $\varepsilon_D \equiv \mathscr{L}(\mathfrak{X}) - \mathscr{L}(X)$. In defining $\overline{\mathscr{L}}$ this way, (1.5) may represent any chosen integration scheme. For implicit schemes, a system would need to be solved [10, 13] and only explicit schemes are considered here. Letting $X_{i+1}^* \in \mathscr{M}_r$ denote the best possible approximation of $\mathfrak{X}_{i+1}$ given information at time t_i, we have

$$X_{i+1}^* = \mathscr{P}_{\mathscr{M}_r}(\mathfrak{X}_i + \Delta t\overline{\mathscr{L}}).$$

Of course, we may not have access to $\mathfrak{X}_i$; we substitute in $X_i \in \mathscr{M}_r$ instead as an approximation.

$$X_{i+1}^* = \mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}) \tag{1.6}$$

Unfortunately, $\mathscr{P}_{\mathscr{M}_r}$, i.e. the truncated SVD, is expensive to compute every time step [59, p. 237]. As such, we seek approximations to the exact projection operator. With the use of (extended) retractions, which map matrices from the embedding Euclidean space back to the manifold, see [1, 12], the numerical scheme then becomes

$$X_{i+1} = \mathcal{R}_{X_i}(\Delta t\overline{\mathscr{L}}), \tag{1.7}$$

where X_{i+1} is no longer the *best* possible approximation given information at time t_i. The difference between (1.6) and (1.7) defines the *projection-retraction error* [12],

$$\varepsilon_{pr} \equiv \mathcal{R}_{X_i}(\Delta t\overline{\mathscr{L}}) - \mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}). \tag{1.8}$$

Using both an integration scheme for $\overline{\mathscr{L}}$ and a retraction of order k, whereby $\varepsilon_{pr} = \mathcal{O}(\Delta t^{k+1})$, guarantees kth-order convergence to the best low-rank approximation assuming that the difference between $X \in \mathscr{M}_r$ and the full-rank system $\mathfrak{X}$ is small. Integration schemes that use projection methods include those presented in [12] as well as projected Runge-Kutta methods [32], which actually use both formulations of the problem. The use of retractions and projection methods is general and may be applied to any integration scheme for $\overline{\mathscr{L}}$.

To wrap up, we note the difference in the derivation of "tangent-space-integration methods," i.e. schemes that integrate (1.2) (e.g. [49, 8, 4, 5]), and projection methods of the form (1.7) (e.g. [25, 26]). To derive (1.2), the Dirac-Frenkel time-dependent variational principle [14, 48] is applied in the continuous time setting, and then time is discretized for numerical integration. For projection methods, there are two approaches, both of which start by discretizing time for $\mathscr{X}$ before making any approximations. The first approach is to integrate the full-rank differential operator $\mathscr{L}(\mathfrak{X}(t),t)$ in the embedding Euclidean space. Then, $\mathfrak{X}$ is approximated as X as in (1.6), and the projection operator is approximated with a retraction as in (1.7). The second approach is to apply an integration scheme to $\mathscr{L}(X(t),t)$; here we integrate along the manifold, so retractions may be applied as we compute $\overline{\mathscr{L}}$. At the last stage, a retraction is applied as in (1.7). So, the difference between the tangent-space-integration and projection methods comes from the order of time discretization for $\mathscr{X}$ and approximation, which manifests in whether or not the tangent-space projection operator $\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}$ is explicitly integrated. Nevertheless, both schemes form consistent integrators.

Progress has been made recently in developing tangent-space-integration methods [49, 8, 4, 5, 30]. Arbitrarily high-order integration schemes may be derived via symmetric composition with the adjoint method. However, this integrator may lack desirable properties of other schemes for particular physical systems. For instance, one may seek a symplectic integrator [51] for Hamiltonian systems, a total-variation diminishing scheme [21] to avoid oscillations in, e.g., shock propagation problems, schemes that conserve relevant physical quantities, or implicit-explicit schemes to preserve stability. This has been studied in model-order reduction [6, 64] and recently in the context of the DLRA [52, 29]. Projection methods allow easy construction of new low-rank integration schemes with our desired features simply by implementing pre-existing full-rank schemes with the addendum of applying a high-order retraction as a final step to project back to the low-rank manifold. One may also use projections/retractions after any given step of a full-rank integration scheme in order to limit the quickly growing rank of $\overline{\mathscr{L}}$ as is done with projected Runge-Kutta methods in [32]. Without a particular choice of gauge, all of these schemes lack mode continuity [61, 16, 47], i.e. the vectors in U should only change slowly as a function of time so that their dynamics are interpretable. Such a feature is essential for the reduced-order evolution of dynamical systems in time, especially for uncertainty quantification. As such, we require efficient and accurate retractions which asymptotically approximate the projection operator and preserve mode continuity as a tool to construct flexible integration schemes.

In [12], we developed a new set of perturbative retractions that approximate the truncated SVD. From these retractions, we created Dynamically Orthogonal Runge-Kutta (DORK) integration schemes, which updated the subspace upon which we projected the system dynamics as we integrated, yielding highly accurate schemes. Nevertheless, integration using these perturbative retractions can have three issues: (i) they can lead to overshooting if time steps are too large, (ii) their higher-order corrections become more expensive as greater accuracy is required, and (iii) they can break down in the presence of small singular values. The tangent-

Notation

ω	Simple event
Ω	Event space
$\mathfrak{X}$	Full-rank system state in $\mathbb{R}^{m\times n}$
X	Low-rank approximation of system state in $\mathscr{M}_r$
$\mathscr{X}$	Either the full-rank $\mathfrak{X}$ or low-rank X system state
χ	Next system state point before retraction, $U_i Z_i^T + \Delta t \overline{\mathscr{L}}$
$\overline{U}$	Time-averaged change in U integrated from t_i to t_{i+1} such that $U_{i+1} = U_i + \Delta t \overline{U}$
$\bar{Z}$	Time-averaged change in Z integrated from t_i to t_{i+1} such that $Z_{i+1} = Z_i + \Delta t \bar{Z}$
$\\| \bullet \\|$	Frobenius norm
$\mathscr{M}_r$	Manifold of rank-r $m \times n$ real matrices
$\overline{\mathscr{L}}$	Approximately or exactly integrated differential operator defined as $\frac{1}{\Delta t}\int_{t_i}^{t_{i+1}} \mathscr{L}(\mathscr{X}(t),t)dt + \mathcal{O}(\Delta t^k)$
$\mathscr{P}_{\mathscr{M}_r}$	Projection onto $\mathscr{M}_r$
$\mathscr{P}_{\mathscr{M}_r}^{\perp}$	Orthogonal complement of projection onto $\mathscr{M}_r$ defined by $I - \mathscr{P}_{\mathscr{M}_r}$
$\mathcal{T}_X\mathscr{M}_r$	Tangent space of low-rank manifold at point X
$\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}$	Projection operator onto tangent space of low-rank manifold at point X
$\mathcal{V}_{m,r}$	Stiefel manifold of $m \times r$ matrices
$\mathcal{V}_{n,r}$	Stiefel manifold of $n \times r$ matrices
$\mathbb{R}_*^{n\times r}$	Set of rank-r matrices in $\mathbb{R}^{n\times r}$
$\mathbb{R}_*^{r\times r}$	Set of rank-r matrices in $\mathbb{R}^{r\times r}$
$\mathcal{U}_{m,r}$	DO space (which depends on U) defined as $\left\{\delta_U \in \mathbb{R}^{m\times r} : U^T\delta_U = 0\right\}$
$so(r)$	Space of skew-symmetric, real $r \times r$ matrices
$\mathcal{T}_U\mathcal{V}_{m,r}$	Tangent space of $m \times r$ Stiefel manifold equivalent to $\left\{\delta_U \in \mathbb{R}^{m\times r} : U^T\delta_U \in so(r)\right\}$
$\mathcal{T}_V\mathcal{V}_{n,r}$	Tangent space of $n \times r$ Stiefel manifold equivalent to $\left\{\delta_V \in \mathbb{R}^{m\times r} : V^T\delta_V \in so(r)\right\}$
$\mathscr{P}_U$	Projection onto modes U defined by UU^T
$\mathscr{P}_V$	Projection onto modes V defined by VV^T
$\mathscr{P}_U^{\perp}$	Orthogonal complement of projection onto modes of U defined by $(I - UU^T)$
$\mathscr{P}_V^{\perp}$	Orthogonal complement of projection onto modes of V defined by $(I - VV^T)$
$\mathcal{R}_A(\Delta t B)$	Retraction from point A in direction B such that $\mathcal{R}_A(\Delta t B) = A + \Delta t \mathscr{P}_{\mathcal{T}_A\mathscr{M}_r}B + \mathcal{O}(\Delta t^2)$
$\mathcal{R}^{(p)}$	Retraction that asymptotically approximates $\mathscr{P}_{\mathscr{M}_r}$ to the p^{th} order.

Table 1 Relevant mathematical structures and operators referenced within the text.

space-integration methods mentioned are robust to small singular values, which enable the recent development of rank-adaptive methods. In [7], the rank of the solution is doubled during the integration of each time step, the solution is integrated, and then the rank is reduced. While this algorithm is effective, it has the computational cost of always computing a solution at double the rank, which is undesirable. There are a few potential algorithmic remedies in the literature to reduce the computational cost by determining when the rank should be increased. Rank-adaptive algorithms proposed in [20, 46] recommend calculating the norm of the residual, namely $\|(I - \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r})\overline{\mathscr{L}}\|$, at each time step; if this value is large, the rank should be augmented. Similarly, [63] recommend using the tangent of the angle between $\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}$ and $\overline{\mathscr{L}}$ as a metric. Rank-adaptive algorithms are also developed in the context of dynamically orthogonal differential equations in [55, 16]. However, the perturbative retractions and DORK schemes in [12] are incompatible with rank adaptation because the rank augmentation necessitates the integration of small singular values which will lead to the inversion of an ill-conditioned correlation matrix $Z^T Z$.

In this paper, we remedy these three issues with a novel set of optimal perturbative retractions. In section 2, we show these retractions are more accurate than those previously introduced at a reduced computational cost and, importantly, largely avoid overshoot. In section 3 we introduce a new simple yet effective set of retractions composed of perturbative

Key expressions

$\{\dot{\mathcal{U}}_j^{\text{orig}}, \dot{\mathcal{Z}}_j^{\text{orig}}\}$	Original first- through fourth-order corrections, see (5.4-5.7) in [12]
$\{\dot{\mathcal{U}}_j\}_{j=1}^4$	Optimal first- through fourth-order corrections (2.6a-2.6d)
U_{i+1}^{robust}	Optimal, robust first-order update to U_i (4.4)
Z_{i+1}^{optimal}	Optimal update to Z_i (4.2)
$\{\dot{\mathcal{U}}_j, \overline{\mathcal{L}}_j\}_{i=1}^4$	First- through fourth-order so-DORK schemes (5.2a-5.2d)
ε_{pr}	Projection-retraction error defined as $\mathcal{R}_X(\Delta t\overline{\mathscr{L}}) - \mathscr{P}_{\mathscr{M}_r}(X + \Delta t\overline{\mathscr{L}})$
ε_l	Local retraction error defined as $\mathcal{R}_X(\Delta t\overline{\mathscr{L}}) - (X + \Delta t\overline{\mathscr{L}})$
$\varepsilon_{\mathcal{N}}$	Normal closure error defined as $\mathscr{P}_{\mathscr{M}_r}^{\perp}(\mathfrak{X}) = \mathfrak{X} - \mathscr{P}_{\mathscr{M}_r}(\mathfrak{X})$
ε_D	Dynamical model closure error defined as $\mathscr{L}(\mathfrak{X}) - \mathscr{L}(X)$
ε_{DF}	Dirac-Frenkel model closure error defined as $\mathscr{L}(\mathfrak{X}) - \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\mathscr{L}(X)$
ε_{tot}	Total error due to dynamical low-rank approximation defined as $\|\mathfrak{X} - X\|$

Table 2 Novel retractions derived and error metrics used throughout this paper. These retractions and error metrics arise due to the dynamic curvature of the low-rank manifold.

retractions, which we call "gradient-descent retractions." Gradient-descent retractions yield exceptional accuracy up to arbitrary order without a quickly growing computational cost. We also develop a simple "automatic" gradient-descent retraction that determines how many iterations to compute based on a user-given tolerance. In section 4, we develop perturbative retractions robust to small singular values that may be coupled with our gradient-descent retractions, and we include rank adaptation as a feature. Then, in section 5 we derive two new low-rank integration schemes based off our new retractions, the stable, optimal Dynamically Orthogonal Runge-Kutta (so-DORK) schemes and gradient-descent Dynamically Orthogonal Runge-Kutta (gd-DORK) schemes. We demonstrate the efficacy of our retractions and integrators with numerical experiments on a matrix differential equation, a linear partial differential equation, and a nonlinear, stochastic partial differential equation in section 6. Finally, we conclude in section 7 with some remarks on the utility of these new retractions and integrators, and we discuss future research directions. Throughout the manuscript, we use the terms "high-order retraction" or pth-order retraction to denote retractions that have projection-retraction error (see table 2) of order $\mathcal{O}(\Delta t^{p+1})$, which is in contrast to a "classical" second-order retraction, whose second-order error belongs to the normal space of $\mathscr{M}_r$.

2. Optimal perturbative retractions.

2.1. Derivation. In this section, we will derive "optimal" perturbative retractions for explicit integration schemes, an improvement over the perturbative retractions developed in [12]. In the original set of perturbative retractions, we approximately solved the following optimization problem.

$$\begin{aligned} &\underset{U_{i+1},Z_{i+1}}{\arg\min} \left\| U_i Z_i^T + \Delta t\overline{\mathscr{L}} - U_{i+1} Z_{i+1}^T \right\| \\ &\text{st } \; U_{i+1} = U_i + \Delta t\vec{U}, \quad \vec{U} \in \mathcal{U}_{m,r} \\ &\quad Z_{i+1} = Z_i + \Delta t\vec{Z} \end{aligned} \tag{2.1}$$

Solving for the first-order optimality conditions of the system yields nonlinear matrix equations, but writing $\vec{U}$ and $\vec{Z}$ as perturbation series in Δt yields linear equations to solve, giving our retractions. For notational convenience, we introduce $\mathcal{X}$,

$$\mathcal{X} \equiv U_i Z_i^T + \Delta t\overline{\mathscr{L}},$$

which denotes the next point we wish to approximate before we retract back to the low-rank manifold (see table 1). Next, we modify the optimization problem (2.1) by solving for Z_{i+1} directly and enforcing numerical orthonormality.

First, suppose that $U_{i+1} \in \mathcal{V}_{m,r}$ is given a priori. In a manner similar to an alternating least squares procedure [58], we can exactly solve for Z_{i+1} in the following unconstrained optimization problem.

$$\underset{Z_{i+1}}{\arg\min} \left\| \mathcal{X} - U_{i+1} Z_{i+1}^T \right\| \quad \Rightarrow \quad Z_{i+1} = \mathcal{X}^T U_{i+1} \tag{2.2}$$

Second, we now would like to substitute this expression for Z_{i+1} in as a constraint in (2.1) to yield a more accurate set of retractions. But, we must first recognize that despite the DO condition, $U_i + \Delta t \vec{U}$ is not exactly orthonormal: $(U_i + \Delta t \vec{U})^T (U_i + \Delta t \vec{U}) = I + \mathcal{O}(\Delta t^2)$. So we cannot simply write $U_{i+1} = U_i + \Delta t \vec{U}$. A re-orthonormalization procedure is typically applied; see [47] for a fast algorithm. This re-orthonormalization amounts to right-multiplying $U_i + \Delta t \vec{U}$ by some nonsingular $r \times r$ matrix, G, such that $G^T (U_i + \Delta t \vec{U})^T (U_i + \Delta t \vec{U}) G = I$. Then, we may let $U_{i+1} = (U_i + \Delta t \vec{U}) G$, and we can rewrite our optimization problem as follows.

$$\begin{aligned} &\underset{U_{i+1}}{\arg\min} \left\| \mathcal{X} - U_{i+1} Z_{i+1}^T \right\| \\ &\text{st } \; U_{i+1} = (U_i + \Delta t \vec{U}) G, \quad \vec{U} \in \mathcal{U}_{m,r} \\ &\quad Z_{i+1} = \mathcal{X}^T U_{i+1}, \quad G \in \mathbb{R}_*^{r \times r} \end{aligned} \tag{2.3}$$

With this formulation, we are now only optimizing over one matrix, U_{i+1} (or $\vec{U}$, equivalently). Additionally, we are ensuring that whatever matrix U_{i+1} we end up with, Z_{i+1} in (2.2) gives the *optimal* low-rank approximation to $\mathcal{X}$. We remark that the update equation for the coefficients is equivalent to that used in proper orthogonal decomposition [50, 3] and other reduced-basis methods such as dynamic mode decomposition [53, 56, 60, 36].

By substituting our constraints into the cost function in (2.3) and differentiating with respect to $U_{i+1} = (U_i + \Delta t \vec{U}) G$, the first-order optimality condition can be simplified to the following nonlinear matrix equation we seek to solve.

$$\begin{aligned} &\frac{\partial}{\partial [(U_i + \Delta t \vec{U}) G]} \left[\left\| \mathcal{X} - (U_i + \Delta t \vec{U}) G G^T (U_i + \Delta t \vec{U})^T \mathcal{X} \right\|^2 \right] = 0 \\ \Leftrightarrow &\left[I - (U_i + \Delta t \vec{U}) G G^T (U_i + \Delta t \vec{U})^T \right] \mathcal{X} \mathcal{X}^T (U_i + \Delta t \vec{U}) = 0 \end{aligned}$$

We may remove the GG^T dependence by using the definition of G to obtain

$$\begin{aligned} GG^T &= \left[(U_i + \Delta t \vec{U})^T (U_i + \Delta t \vec{U}) \right]^{-1} \\ &= \left[I + \Delta t^2 \vec{U}^T \vec{U} \right]^{-1} \\ &= I - \Delta t^2 \vec{U}^T \vec{U} + \Delta t^4 (\vec{U}^T \vec{U})^2 - \cdots . \end{aligned}$$

The last line follows from the Neumann series of the inverse of $I + A$ for some matrix A, which converges assuming that $\Delta t^2 \| \vec{U}^T \vec{U} \| < 1$; this is not a restrictive assumption since Δt can always be chosen to be smaller such that this holds, and this is further remedied by our robust retractions in section 4. Note that since G has been eliminated from our equation, it need not be explicitly computed but will be computed implicitly when re-orthonormalizing U_{i+1}. Then our optimality condition becomes the following.

$$\left[I - (U_i + \Delta t \vec{U})(I - \Delta t^2 \vec{U}^T \vec{U} + \cdots)(U_i + \Delta t \vec{U})^T \right] \mathcal{X} \mathcal{X}^T (U_i + \Delta t \vec{U}) = 0 \tag{2.4}$$

Now, we assume that $\vec{U}$ may be written as a perturbation series in Δt,

$$\Delta t \vec{U} = \sum_{j=1}^{\infty} \Delta t^j \dot{\mathcal{U}}_j . \tag{2.5}$$

After substituting (2.5) into (2.4), we may solve for $\dot{\mathcal{U}}_j$ via small linear systems by grouping terms of the same order of Δt. We refer to [9, 12] for an analogous derivation. Below, we provide first- through fourth-order corrections.

$$\dot{\mathcal{U}}_1 = \mathscr{P}^{\perp}_{U_i}\overline{\mathscr{L}}Z_i(Z_i^TZ_i)^{-1} \tag{2.6a}$$

$$\dot{\mathcal{U}}_2 = \left[\mathscr{P}^{\perp}_{U_i}\overline{\mathscr{L}\mathscr{L}^T}U_i - \dot{\mathcal{U}}_1\left(U_i^T\overline{\mathscr{L}}Z_i + Z_i^T\overline{\mathscr{L}}^TU_i\right)\right](Z_i^TZ_i)^{-1} \tag{2.6b}$$

$$\begin{aligned}\dot{\mathcal{U}}_3 = \Big[&\mathscr{P}^{\perp}_{U_i}\overline{\mathscr{L}\mathscr{L}^T}\dot{\mathcal{U}}_1 - \dot{\mathcal{U}}_2\left(U_i^T\overline{\mathscr{L}}Z_i + Z_i^T\overline{\mathscr{L}}^TU_i\right)\\ &- \dot{\mathcal{U}}_1\left(U_i^T\overline{\mathscr{L}\mathscr{L}^T}U_i + \dot{\mathcal{U}}_1^T\overline{\mathscr{L}}Z_i + Z_i^T\overline{\mathscr{L}}^T\dot{\mathcal{U}}_1 - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1Z_i^TZ_i\right)\Big](Z_i^TZ_i)^{-1}\end{aligned} \tag{2.6c}$$

$$\begin{aligned}\dot{\mathcal{U}}_4 = \Big[&\mathscr{P}^{\perp}_{U_i}\overline{\mathscr{L}\mathscr{L}^T}\dot{\mathcal{U}}_2 - \dot{\mathcal{U}}_3\left(U_i^T\overline{\mathscr{L}}Z_i + Z_i^T\overline{\mathscr{L}}^TU_i\right)\\ &- \dot{\mathcal{U}}_2\left(U_i^T\overline{\mathscr{L}\mathscr{L}^T}U_i + \dot{\mathcal{U}}_1^T\overline{\mathscr{L}}Z_i + Z_i^T\overline{\mathscr{L}}^T\dot{\mathcal{U}}_1 - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1Z_i^TZ_i\right)\\ &- \dot{\mathcal{U}}_1\Big(U_i^T\overline{\mathscr{L}\mathscr{L}^T}\dot{\mathcal{U}}_1 + \dot{\mathcal{U}}_1\overline{\mathscr{L}\mathscr{L}^T}U_i + \dot{\mathcal{U}}_2^T\overline{\mathscr{L}}Z_i + Z_i^T\overline{\mathscr{L}}^T\dot{\mathcal{U}}_2 - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_2Z_i^TZ_i\\ &- \dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_1Z_i^TZ_i - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1U_i^T\overline{\mathscr{L}}Z_i - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1Z_i^T\overline{\mathscr{L}}^TU_i\Big)\Big](Z_i^TZ_i)^{-1}\end{aligned} \tag{2.6d}$$

We note here that there are two patterns to exploit for computational efficiency in (2.6a-2.6d). First, the high-order corrections require the lower-order corrections, and each term on the right-hand side that is left-multiplied by $\dot{\mathcal{U}}_j$ is repeated in the higher-order corrections. So, the repeated terms should be stored to avoid extra matrix multiplications. Second, several of the terms above are transposes of each other, so again, storage will avoid redundant computation.

For concreteness, we explicitly write out a second-order retraction with our optimal perturbative retraction.

$$\begin{gathered}X_{i+1} = \mathcal{R}^{\text{opt-2}}_{X_i}(\Delta t\overline{\mathscr{L}}) = U_{i+1}Z_{i+1}^T\\ U_{i+1} = \texttt{orth}\left(U_i + \Delta t\,\dot{\mathcal{U}}_1 + \Delta t^2\,\dot{\mathcal{U}}_2\right), \qquad Z_{i+1} = (X_i + \Delta t\overline{\mathscr{L}})^TU_{i+1}\end{gathered} \tag{2.7}$$

The first- and second-order corrections, $\dot{\mathcal{U}}_1$ and $\dot{\mathcal{U}}_2$, respectively, are given by equations (2.6a) and (2.6b), and the orthonormalization procedure — which implicitly generates the G matrix referenced beforehand — may be a QR decomposition or other fast algorithm as in [47].

It is instructive to derive the *local retraction error*, ε_l, of (i) any generic retraction and (ii) our optimal pertubative retractions in (2.6a-2.6d). From the generic definition of a retraction in Table 1, any local retraction error may be written as

$$\begin{aligned}\varepsilon_l &\equiv X_i + \Delta t\overline{\mathscr{L}} - \mathcal{R}_{X_i}(\Delta t\overline{\mathscr{L}})\\ &= U_iZ_i^T + \Delta t\overline{\mathscr{L}} - \left(U_iZ_i^T + \Delta t\,\mathscr{P}_{\mathcal{T}_{X_i}\mathscr{M}_r}\overline{\mathscr{L}} + \mathcal{O}(\Delta t^2)\right)\\ &= \left(I - \mathscr{P}_{\mathcal{T}_{X_i}\mathscr{M}_r}\right)\Delta t\overline{\mathscr{L}} + \mathcal{O}(\Delta t^2).\end{aligned}$$

Explicitly substituting in our expression for $\mathscr{P}_{\mathcal{T}_{X_i}\mathscr{M}_r}$ (see [17]), we obtain

$$\varepsilon_l = (I - U_iU_i^T)\Delta t\overline{\mathscr{L}}(I - Z_i(Z_i^TZ_i)^{-1}Z_i^T) + \mathcal{O}(\Delta t^2) \tag{2.8}$$

For our optimal perturbative retractions, we may write the local retraction error as

$$\varepsilon_l^{\text{opt}} = U_iZ_i^T + \Delta t\overline{\mathscr{L}} - U_{i+1}Z_{i+1}^T,$$

noting that $U_{i+1}Z_{i+1}^T = \mathcal{R}^{\text{opt}}_{X_i}(\Delta t\overline{\mathscr{L}})$ by definition. Now, we substitute in our expression for Z_{i+1} from equation (2.2).

$$\begin{aligned}\varepsilon_l^{\text{opt}} &= U_iZ_i^T + \Delta t\overline{\mathscr{L}} - U_{i+1}U_{i+1}^T(U_iZ_i^T + \Delta t\overline{\mathscr{L}})\\ &= (I - U_{i+1}U_{i+1}^T)\chi\end{aligned} \tag{2.9}$$

Contrasting the generic retraction error (2.8) with the optimal perturbative local retraction error (2.9), we see that our particular choice of Z_{i+1} has allowed us to write a cleaner, more interpretable form of ε_l. By insisting that Z_{i+1} is optimal given U_{i+1}, our error is given as the U_{i+1} orthogonal projection of our new point. In words, our error is given by the information not captured by the span of the to-be-computed U_{i+1}. The best U_{i+1} would then be the r left singular vectors of χ with the largest singular values, but, of course, this would be expensive to compute. Because the truncated SVD is the exact solution to (2.3) which our retractions approximately solve, the left singular vectors are precisely what our perturbation series asymptotically approaches.

Lastly, we consider the stability of the retractions.

LEMMA 2.1. *Let $X_{i+1} = \mathcal{R}^{opt}_{X_i}(\Delta t\overline{\mathscr{L}})$ where we use the optimal perturbative retractions given by equations (2.6a-2.6d) and (2.2). Then $\|X_{i+1}\| \leq \|X_i + \Delta t\overline{\mathscr{L}}\|$.*

Proof. Let $X_{i+1} = U_{i+1}Z^T_{i+1}$. Using the optimality (2.2), $X_{i+1} = U_{i+1}U^T_{i+1}(X_i + \Delta t\overline{\mathscr{L}})$. By construction, $U_{i+1} \in \mathcal{V}_{m,r}$, so $\|X_{i+1}\| \leq \|X_i + \Delta t\overline{\mathscr{L}}\|$. □

We remark that this stability property is unique to the *optimal* perturbative retractions as it relies on the coefficients Z_{i+1} being calculated as the projection of the next state onto a semi-orthonormal basis via equation (2.2). This is in contrast to the original perturbative retractions in which the Z_{i+1}'s had their own perturbation series.

2.2. Illustration. We now exemplify the projection-retraction error of the optimal perturbative retractions (see eqs. (2.5, 2.6) and Table 2). For comparison, we include the error from the original adaptive perturbative algorithm (using a hyperparameter $\varepsilon = 0.1$) from [12] which automatically determines the best order of retraction to apply. Overshoot typically occurs with the original high-order perturbative retractions [12] when Δt is large because the perturbation series does not converge quickly. From another point of view, these high-order perturbations project the dynamics onto polynomial approximations of the low-rank manifold, but such polynomial approximations grow inaccurate for large steps. To avoid overshoot and ensure the perturbation series does not diverge catastrophically, the adaptive algorithm measures the norm of the high-order corrections and only includes terms that have a norm less than some user-given ε. Though both the original and optimal perturbative retractions orthonormalize the modes after each time step (and hence $\|U_{i+1}\| = 1$), only the optimal perturbative retractions prevent the coefficients Z_{i+1} from growing too large. That is, our optimal retractions tend to avoid this overshoot (or at least limit it) due to their optimal choice of coefficients. This choice of coefficients is made by possible without pseudoinversion due to the re-orthonormalization procedure of U_{i+1}.

The example problem of matrix addition is constructed as follows. We approximate $X_0 + \Delta t\overline{\mathscr{L}}$ by forming our rank-10 starting point $X_0 \in \mathbb{R}^{500\times 220}$; we choose uniformly randomly sampled matrices $U_0 \in \mathbb{R}^{500\times 10}$ and $V_0 \in \mathbb{R}^{220\times 10}$, and then we orthonormalize them. Then, we choose $S_0 \in \mathbb{R}^{10\times 10}$, also uniformly randomly sampled, and normalize S_0 such that its Frobenius norm is one. Finally, we set $X_0 = U_0 S_0 V_0^T$. Next, we form a rank-100 matrix $\overline{\mathscr{L}} \in \mathbb{R}^{500\times 220}$ which is constructed by multiplying randomly, uniformly sampled matrices of size 500 × 100 and 100 × 220, and then we normalize $\overline{\mathscr{L}}$ to have Frobenius norm one. Our reference solution then becomes $\mathscr{P}_{\mathscr{M}_r}(X_0 + \Delta t\overline{\mathscr{L}})$, and we compare that with $\mathcal{R}_{X_0}(\Delta t\overline{\mathscr{L}})$.

We illustrate the results in Figures 1 and 2. We observe that the kth-order perturbative retraction exhibits linear convergence to the best low-rank approximation at order $k+1$, where the $+1$ comes from only analyzing the local error instead of the global error after $\mathcal{O}(1/\Delta t)$ steps of time integration. In addition, the optimal perturbative retractions are essentially always better than the original perturbative retractions.

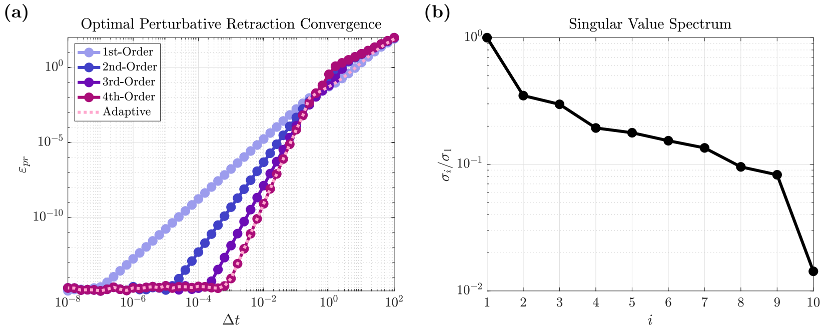

Fig. 1: (a) Our optimal perturbative retractions (2.6), e.g. (2.7), exhibit high-order convergence to the best low-rank approximation. We see linear convergence at rates 2, 3, 4, and 5 in ε_{pr} (Table 2). The adaptive algorithm selects which order correction to use by analyzing the convergence of the perturbation series. (b) The singular value spectrum of X_0, showing singular values span nearly two orders of magnitude. The condition number of $S_0^T S_0$ in the Frobenius norm is 4968.3.

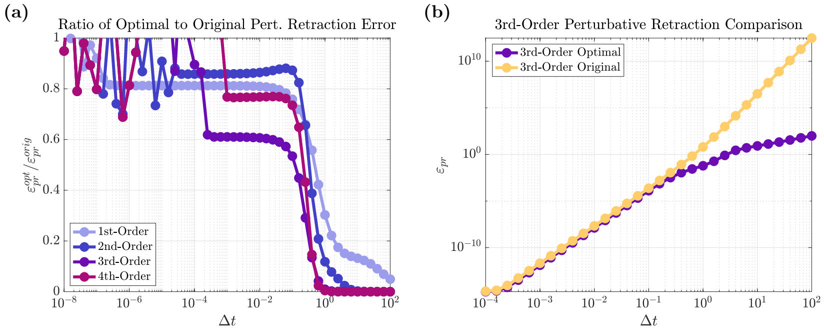

Fig. 2: (a) Ratio of errors of the optimal (2.6), e.g. (2.7), and original [12] perturbative retractions. The ratio is less than one, indicating the optimal retractions offer an improvement until Δt becomes small enough that both retractions' errors are near machine zero, so the ratio fluctuates around one. (b) Explicit comparison of the errors between third-order retractions. The new optimal retraction (2.6c) is always better, as expected, but performs especially well with large Δt by mostly avoiding overshoot [12].

3. Gradient-descent retractions.

3.1. Derivation. In this section, we demonstrate how to accelerate convergence to the best low-rank approximation by iteratively applying pre-existing retractions. This methodology generates what we call "gradient-descent retractions," and we show that if the retractions asymptotically approximate the exact projection operator, the gradient-descent retractions converge *superlinearly* to a low-rank matrix $X \in \mathscr{M}_r$ and *linearly* to the best low-rank approximation of a matrix $X \notin \mathscr{M}_r$.

We begin by recalling the geometric intuition behind Newton's method for minimizing some cost function $J(x)$ in one dimension. By expanding $J(x_k + \Delta x)$ around an iterate x_k, we obtain a second-order Taylor series $J(x_k + \Delta x) \approx J(x_k) + \Delta x J'(x_k) + \frac{\Delta x^2}{2} J''(x_k)$. To minimize $J(x_k + \Delta x)$, we differentiate our expression with respect to Δx and set it to zero, giving $\Delta x = -J''(x_k)^{-1} J'(x_k)$. In words, we minimized a second-order approximation of the cost function J. Alternatively, we note that the first-order optimality constraint is given by $J'(x) = 0$. Newton's method for solving this potentially nonlinear equation is given by Taylor expanding J' in a similar fashion: $J'(x_k + \Delta x) \approx J'(x_k) + \Delta x J''(x_k)$. We set this first-order expression to zero as insisted by our optimality constraint, which yields the same expression for Δx. In this interpretation, we solve for the zero of a linear approximation of J' [41].

Now, consider our optimization problem (2.3) with optimality condition (2.4). By expanding $\Delta t \vec{U}$ as a perturbation series, implicitly, we are deriving the Taylor series of (2.4). We may interpret the first-order perturbative retraction as applying one Newton step. Furthermore, our higher-order perturbative retractions may be interpreted then as one step of higher-order gradient-descent iterations generalizing Newton's method. Denoting $\mathcal{X} - X_i$ as the residual between the new point of arbitrary rank and our current point, this interpretation leads to a new iterative algorithm for improving these retractions: simply repeat the retractions applied on the residual, just as one iterates in Newton's method. Such a gradient-descent retraction scheme is defined in algorithm 3.1. It is also natural to consider a retraction where the number of iterations is determined automatically, iterating until some convergence criterion is met. We suggest measuring the Frobenius norm of the difference between iterates of X_{i+1} normalized by the norm of X_i; when the norm of the difference drops below a pre-specified value/tolerance, denoted by hyperparameter Δ^*, or we exceed a certain number iterations, denoted by $N_{\max}$, we stop iterating. An automatic gradient-descent retraction is written out in algorithm 3.2.

Algorithm 3.1 Fixed-order gradient-descent retraction

```
Input: $X_i \in \mathscr{M}_r$, $\overline{\mathscr{L}} \in \mathbb{R}^{m \times n}$, $\Delta t \in \mathbb{R}$, $N \in \mathbb{Z}^+$
Output: $X_{i+1} \in \mathscr{M}_r$
$X_{i+1} = X_i$
for $j = 1 : N$ do
    $X_{i+1} \leftarrow \mathcal{R}_{X_{i+1}}(\Delta t \overline{\mathscr{L}} - (X_{i+1} - X_i))$
end for
```

Algorithm 3.2 Automatic gradient-descent retraction

```
Input: $X_i \in \mathscr{M}_r$, $\overline{\mathscr{L}} \in \mathbb{R}^{m \times n}$, $\Delta t \in \mathbb{R}$, $N_{\max} \in \mathbb{Z}^+$, $\Delta^* \in \mathbb{R}$
Output: $X_{i+1} \in \mathscr{M}_r$
$\alpha = \|X_i\|$
$X_{i+1} = X_i$
$X_{i+1} \leftarrow \mathcal{R}_{X_{i+1}}(\Delta t \overline{\mathscr{L}} - (X_{i+1} - X_i))$
$\Delta = \|X_{i+1} - X_i\|$
$j = 1$
while $j < N_{\max}$ and $\Delta / \alpha < \Delta^*$ do
    $j \leftarrow j + 1$
    $\hat{X} = X_{i+1}$
    $X_{i+1} \leftarrow \mathcal{R}_{X_{i+1}}(\Delta t \overline{\mathscr{L}} - (X_{i+1} - X_i))$
    $\Delta = \|X_{i+1} - \hat{X}\|$
end while
```

Recall that in Algorithms 3.1 and 3.2, whenever X_{i+1} is updated via a retraction, U_{i+1} and Z_{i+1} are updated since X_{i+1} is just shorthand for $U_{i+1} Z_{i+1}^T$. In fact, X_{i+1} is never formed, and doing so may not even be possible on a memory-limited device. In our implementation, we find that using a first-order optimal perturbative retraction (2.2, 2.6a) or its robust variant (4.2, 4.4) (to be discussed in section 4.1.1) is efficient and extends stability from lemma 2.1, which will be shown next.

Finally, as we consider explicit schemes in this work, we remark that $\overline{\mathscr{L}}$ can depend on the past iteration points. This approach is used with perturbations to derive DORK schemes in section 5. In general, $\overline{\mathscr{L}}$ may also be computed with iterative semi-implicit or implicit schemes and then depend on unknown or future points within Δt [10, 13].

3.2. Analysis. First, we show that the gradient-descent retractions are stable provided that the inner retraction chosen —- such as an optimal perturbative retraction — is also stable.

LEMMA 3.1. *Let $X_{i+1} = \mathcal{R}^{gd}_{X_i}\left(\Delta t\overline{\mathscr{L}}\right)$ be the result of applying a gradient-descent retraction from algorithm 3.1 or 3.2 using an optimal retraction $\mathcal{R}^{opt}$ given by equations (2.6a-2.6d) and (2.2) as the final inner retraction. Then $\|X_{i+1}\| \leq \|X_i + \Delta t\overline{\mathscr{L}}\|$.*

Proof. Assume we perform k inner iterations of gradient descent. From the last iteration of the gradient-descent retraction, we have that

$$\begin{aligned}\left\|X^{(k)}_{i+1}\right\| &= \left\|\mathcal{R}^{\text{opt}}_{X^{(k-1)}_{i+1}}\left(X_i + \Delta t\overline{\mathscr{L}} - X^{(k-1)}_{i+1}\right)\right\| \\ &\leq \|X^{(k-1)}_{i+1} + X_i + \Delta t\overline{\mathscr{L}} - X^{(k-1)}_{i+1}\| \\ &= \|X_i + \Delta t\overline{\mathscr{L}}\|\end{aligned}$$

The inequality above follows from lemma 2.1. □

Hence using the stable, optimal retractions in our gradient-descent retractions retains our stability property. We emphasize that only the final inner retraction in the gradient descent iterations must be stable in order to preserve the overall stability of the retractions.

Next, we analyze the rate of convergence of Algorithm 3.1 by first assuming that the point we wish to approximate, $X_i + \Delta t\overline{\mathscr{L}}$, is rank-$r$ and then analyzing the case where it is not, i.e., $X_i + \Delta t\overline{\mathscr{L}} \notin \mathscr{M}_r$. Recall that throughout the text, we assume that $\overline{\mathscr{L}}$ is computed explicitly, meaning it may depend on previous states including X_{i-1} and X_i but not unknown or future states such as X_{i+1}. Letting r_L denote the rank of $\overline{\mathscr{L}}$, we remark that even if $X_i + \Delta t\overline{\mathscr{L}} \in \mathscr{M}_r$, naively computing the sum (without retracting) will result in storing $U_{i+1} \in \mathbb{R}^{m\times(r+r_L)}$ and $Z_{i+1} \in \mathbb{R}^{n\times(r+r_L)}$ in contrast to $U_{i+1} \in \mathbb{R}^{m\times r}$ and $Z_{i+1} \in \mathbb{R}^{n\times r}$ after retracting back to $\mathscr{M}_r$. Algorithm 3.1 provides an efficient way to limit the growth of the rank of X after each time step.

LEMMA 3.2. *If $X_i + \Delta t\overline{\mathscr{L}} \in \mathscr{M}_r$, for a small enough Δt such that (3.4) holds, iteratively applying retractions that asymptotically approximate $\mathscr{P}_{\mathscr{M}_r}$ to the p$^{\text{th}}$ order ($\varepsilon_{pr} = \mathcal{O}(\Delta t^p)$) yields rate-p superlinear convergence to the fixed point $X_i + \Delta t\overline{\mathscr{L}}$.*

Proof. We wish to approximate some point $X_i + \Delta t\overline{\mathscr{L}}$, and we define an iteration of the following form.

$$X^{(j)} = \mathcal{R}^{(p)}_{X^{(j-1)}}(X_i + \Delta t\overline{\mathscr{L}} - X^{(j-1)}) \tag{3.1}$$

Above, $\mathcal{R}^{(p)}$ denotes a retraction that asymptotically approximates $\mathscr{P}_{\mathscr{M}_r}$ to the p^{th} order, and j is our iteration index. By definition, we then have that

$$\mathcal{R}^{(p)}_{X^{(j-1)}}(X_i + \Delta t\overline{\mathscr{L}} - X^{(j-1)}) = \mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}) + \mathcal{O}\left(\left\|X_i + \Delta t\overline{\mathscr{L}} - X^{(j-1)}\right\|^p\right). \tag{3.2}$$

Note that $\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}})$ is the fixed point. Let $\varepsilon^{(j)}_{pr} = X^{(j)} - \mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}})$ denote the projection-retraction error at iteration j. We may write $\|\varepsilon^{(j)}_{pr}\|$ as a function of $\varepsilon^{(j-1)}_{pr}$ by combining (3.1) and (3.2).

$$\begin{aligned}\left\|\varepsilon^{(j)}_{pr}\right\| &= \mathcal{O}\left(\left\|X_i + \Delta t\overline{\mathscr{L}} - X^{(j-1)}\right\|^p\right) \\ &= \mathcal{O}\left(\left\|\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}) - X^{(j-1)}\right\|^p\right) \\ &= \mathcal{O}\left(\left\|\varepsilon^{(j-1)}_{pr}\right\|^p\right)\end{aligned}$$

Above, we have used the fact that $X_i + \Delta t\overline{\mathscr{L}} \in \mathscr{M}_r$. Hence, we have shown the iteration converges superlinearly at rate p assuming a close enough starting point is chosen such that the sequence converges. □

Note that a perturbative retraction of order $p-1$ may be used as $\mathcal{R}^{(p)}$ since it asymptotically approximates the solution to (2.1) and thus asymptotically approximates $\mathscr{P}_{\mathscr{M}_r}$ to the p^{th} order.

If $X_i + \Delta t\overline{\mathscr{L}} \notin \mathscr{M}_r$, as a first remark, we can expect to see at least linear convergence to the best low-rank approximation by iteratively applying smooth retractions; that is, assuming the iteration converges at all. To see why, consider a new function g.

$$g(X) \equiv \mathcal{R}_X(X_i + \Delta t\overline{\mathscr{L}} - X)$$

Then, we may expand $g(X^{(j-1)})$ at $\mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}\right)$ via Taylor series.

$$\begin{aligned} g(X^{(j-1)}) &= g\left(\mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}\right)\right) \\ &\quad + \left\langle \varepsilon_{pr}^{(j-1)}, g'\left(\mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}\right)\right)\right\rangle + \mathcal{O}\left(\left\|\varepsilon_{pr}^{(j-1)}\right\|^2\right) \end{aligned} \tag{3.3}$$

This assumes our retractions vary smoothly, which is generally valid except perhaps at singular value crossings (see [17]). We note that $g(\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}})) = \mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}})$, and the left-hand side of (3.3) is equal to $X^{(j)}$. Subtracting $\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}})$ from both sides gives us $\varepsilon_{pr}^{(j)}$, and so we have the following.

$$\left\|\varepsilon_{pr}^{(j)}\right\| = \mathcal{O}\left(\left\|\varepsilon_{pr}^{(j-1)}\right\|\right)$$

We have shown that if the iteration converges, it at least converges linearly.

To show the iteration converges using the above, we would need to show that $\|g'(\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}))\| < 1$, but this is dependent upon the retraction chosen. However, we can instead derive a sufficient condition for convergence. Starting from (3.2), again without assuming $X_i + \Delta t\overline{\mathscr{L}} \in \mathscr{M}_r$, we have that $\left\|\varepsilon_{pr}^{(j)}\right\| = \mathcal{O}\left(\left\|X_i + \Delta t\overline{\mathscr{L}} - X^{(j-1)}\right\|^p\right)$ and thus

$$\left\|\varepsilon_{pr}^{(j)}\right\| \leq \alpha \left\|X_i + \Delta t\overline{\mathscr{L}} - X^{(j-1)}\right\|^p$$

for some $\alpha \in \mathbb{R}$. We can then break the error into different components, denoting the orthogonal complement of the projection operator $\mathscr{P}_{\mathscr{M}_r}^{\perp} \equiv I - \mathscr{P}_{\mathscr{M}_r}$.

$$\begin{aligned} \left\|\varepsilon_{pr}^{(j)}\right\| &\leq \alpha \left\|\mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}\right) + \mathscr{P}_{\mathscr{M}_r}^{\perp}\left(X_i + \Delta t\overline{\mathscr{L}}\right) - X^{(j-1)}\right\|^p \\ &= \alpha \left\|\mathscr{P}_{\mathscr{M}_r}^{\perp}\left(X_i + \Delta t\overline{\mathscr{L}}\right) - \varepsilon_{pr}^{(j-1)}\right\|^p \\ &\leq \alpha \left(\left\|\varepsilon_{pr}^{(j-1)}\right\| + \left\|\mathscr{P}_{\mathscr{M}_r}^{\perp}\left(X_i + \Delta t\overline{\mathscr{L}}\right)\right\|\right)^p \end{aligned}$$

Recall that $\mathscr{P}_{\mathscr{M}_r}^{\perp}\left(X_i + \Delta t\overline{\mathscr{L}}\right)$ is related to the normal closure error $\varepsilon_{\mathcal{N}}$, the distance between the best low-rank approximation and $X_i + \Delta t\overline{\mathscr{L}}$. We make the assumption that this value is small, otherwise, the low-rank approximation would be inaccurate. So, it is reasonable to insist that

$$\left\|\mathscr{P}_{\mathscr{M}_r}^{\perp}\left(X_i + \Delta t\overline{\mathscr{L}}\right)\right\| < \left(\frac{\left\|\varepsilon_{pr}^{(j-1)}\right\|}{\alpha}\right)^{\frac{1}{p}} - \left\|\varepsilon_{pr}^{(j-1)}\right\|, \tag{3.4}$$

where $\|\varepsilon_{pr}^{(j-1)}\|$ can also be made arbitrarily small by choosing a point sufficiently close to $\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}})$. This can easily be accomplished, no matter the dynamics $\overline{\mathscr{L}}$, by refining the time step Δt. Hence, if (3.4) is satisfied (e.g., $\varepsilon_{\mathcal{N}}$ decays fast enough with Δt), our iteration converges since we then have shown $\|\varepsilon_{pr}^{(j)}\| < \|\varepsilon_{pr}^{(j-1)}\|$. We interpret this sufficient condition by considering the singular values of the quantities in (3.4). Specifically, $\|\varepsilon_{pr}^{(j-1)}\| = \mathcal{O}(\sigma_1^{(j-1)})$ where $\sigma_1^{(j-1)}$ denotes the largest singular value of $\varepsilon_{pr}^{(j-1)}$, i.e. the dominant dynamics not yet captured in $X^{(j-1)}$, and $\|\mathscr{P}_{\mathscr{M}_r}^{\perp}\left(X_i + \Delta t\overline{\mathscr{L}}\right)\| = \mathcal{O}(\sigma_{r+1}^{(i+1)})$, where $\sigma_{r+1}^{(i+1)}$ is the largest singular value of the projection of the dynamics over Δt in the orthogonal complement space (recall X_i is of rank r). As long as $\sigma_{r+1}^{i+1} \ll \sigma_m^{(j-1)}$, the iteration will converge; if not, the total error will be dominated by the error in the normal space (see figure 5). For an in-depth analysis of the decay of this $\mathscr{P}_{\mathscr{M}_r}^{\perp}\left(X_i + \Delta t\overline{\mathscr{L}}\right)$, we refer to [34]. In summary, high-order convergence is preserved if the initial point is close enough to the fixed point, and that distance is less than the smallest singular value of the fixed point. If the smallest singular value of the fixed point is negligible, e.g., of the order of machine precision, the rank of the system should be truncated, which is discussed in section 4.1.3.

3.3. Illustration. To demonstrate the efficacy of the gradient-descent and optimal perturbative retractions, we study their convergence properties. We use the same problem setup as in Section 2.2, hence $i = 0$.

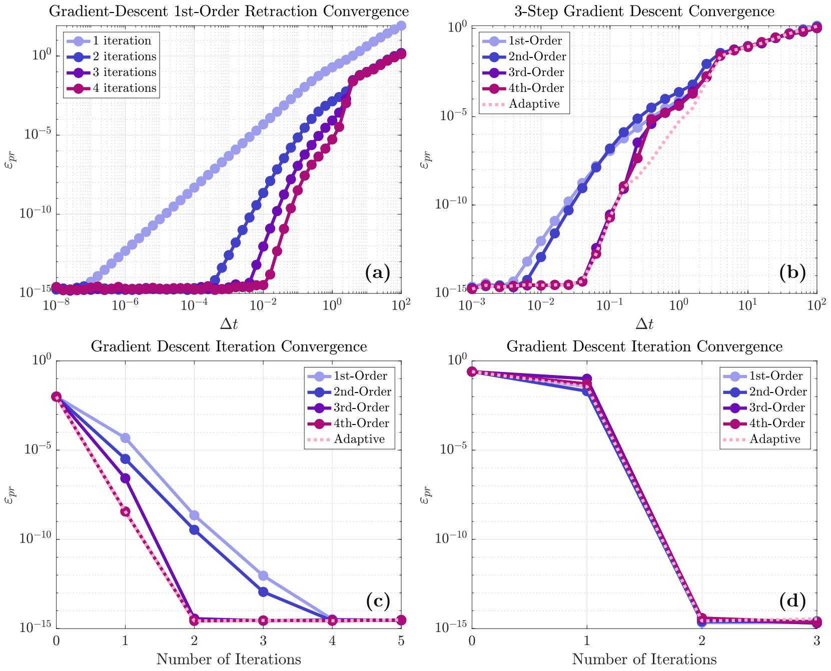

Fig. 3: (a) Convergence of the gradient-descent iterations with various Δt using a first-order perturbative retraction (2.6a). Using more iterations provides more accurate approximations. (b) Convergence using 3 steps of gradient descent with different retractions; high-order (and adaptive-order [12]) retractions lead to faster convergence. (c) Projection-retraction error vs. iteration number using different retractions with $\Delta t = 0.01$. Again, the higher-order retractions converge the fastest. (d) Projection-retraction error vs. iteration number with $\Delta t = 0.25$, but here we insist $X_0 + \Delta t\overline{\mathscr{L}}' \in \mathscr{M}_r$. All retractions converge to the best approximation to machine precision in two iterations.

In Figure 3, the first three panels are analogous to those of figures 1 and 2, where $X_0 + \Delta t\overline{\mathscr{L}} \notin \mathscr{M}_r$. By just using first-order retractions iteratively, we obtain high-order convergence to the best low-rank approximation. Hence, the computational cost for high-order retractions does not grow due to the larger expressions for higher-order corrections. In addition, we observe that utilizing higher-order perturbative retractions with multiple iterations accelerates convergence further. In the bottom right panel, we swap out $\overline{\mathscr{L}}$ for $\overline{\mathscr{L}}'$ such that $X_0 + \Delta t\overline{\mathscr{L}}' =$

$\mathscr{P}_{\mathscr{M}_r}(X_0 + \Delta t \overline{\mathscr{L}})$. In this case, even with a relatively large Δt, we observe extremely rapid convergence to the best low-rank approximation, which suggests (but does not rigorously show) superlinear convergence.

4. Robust retractions with rank adaptation.

4.1. Derivation.

So far, we have shown how to approximate the truncated SVD via a perturbation series, yielding novel, effective retractions accurate up to arbitrary order. We have also shown how to iterate on the low-rank manifold to perform arbitrarily-high-order gradient descent on the low-rank manifold using these retractions. In this section, we develop an algorithm to perform rank-adaptive integration, which requires a retraction that is robust to small singular values.

To see why a robust retraction for rank-adaptive integration is needed, consider the first step at time t_{i+1} after augmenting the rank of our system from rank r_i to r_{i+1}. We denote the singular values of X_i as $\sigma_i^{(j)}$, $j = 1, \ldots, r_i$, and those of X_{i+1} as $\sigma_{i+1}^{(j)}$, $j = 1, \ldots, r_{i+1}$. By equation (1.6), for the best possible approximation $X_{i+1}^* = \mathscr{P}_{\mathscr{M}_{r_{i+1}}}(X_i + \Delta t \overline{\mathscr{L}})$, we then have by Weyl's inequality [62],

$$\sigma_{i+1}^{(j)} \leq \sigma_i^{(j)} + \mathcal{O}(\Delta t).$$

This, of course, assumes that $\|\overline{\mathscr{L}}\| = \mathcal{O}(1)$. The implications are that for $r_i < j \leq r_{i+1}$, as $\sigma_i^{(j)} = 0$, we have $\sigma_{i+1}^{(j)} \sim \Delta t$, and so X_{i+1} will be ill-conditioned assuming $\|X_{i+1}\| = \mathcal{O}(1)$. As such, we must avoid inverting $Z_i^T Z_i$ in our retractions.

4.1.1. Robust retractions.

Here, we develop two methods to robustify our high-order retractions. The most straightforward approach is as follows: instead of inverting $Z_i^T Z_i$, we simply use the Moore-Penrose pseudoinverse. When $Z_i^T Z_i$ is rank-deficient, the linear systems (2.6a-2.6d) do not admit unique solutions for $\{\dot{\mathcal{U}}_j\}_j$. As such, we may regularize the system and still satisfy the equations. So that the perturbation series converges, it is natural to find least-squares solutions, which is precisely what the pseudoinverse does. The pseudoinverse of any given matrix A may be calculated by taking the truncated SVD (or eigendecomposition), USV^T, and then inverting singular values with magnitude greater than some threshold. Since we always invert a Hermitian matrix, a more numerically accurate algorithm takes the truncated SVD of A and then the QR decomposition of $\sqrt{S}U^T$; the transpose of the resulting upper triangular matrix R gives a Cholesky decomposition of $A = R^T R$. To solve a rank-r linear system, we can then simply use forward and backward substitution. Otherwise, in the rank-deficient case, a linear least-squares problem is solved.

The advantages of this pseudoinverse approach are that it can be used with both our high-order perturbative retractions and incorporated into our gradient-descent retractions, and it preserves mode continuity, which is desirable for uncertainty quantification [46, 47]. The pseudoinversion, however, induces an additional source of error accumulation. When the singular values of X_i are small but nonzero, truncating the singular values of the correlation matrix is inherently another approximation, and the truncation threshold we choose generates a lower bound on the minimum error a low-rank integration scheme can attain. This effect is depicted in Section 6, Fig. 6.

To avoid the error accumulation due to the pseudoinverse, we introduce an alternative robustification technique by building off of our first-order retraction (2.6a) before re-orthonormalization.

$$U_{i+1} = U_i + \Delta t \mathscr{P}_{U_i}^{\perp} \overline{\mathscr{L}} Z_i \left(Z_i^T Z_i\right)^{-1} \tag{4.1}$$

$$Z_{i+1}^{\text{optimal}} = Z_i U_i^T U_{i+1} + \Delta t \overline{\mathscr{L}}^T U_{i+1} \tag{4.2}$$

Importantly, we only care about the column space of U_{i+1} rather than its individual columns because Z_{i+1} will optimally adapt to any rotation to U_{i+1}. So, we right-multiply our update equation for U by $Z_i^T Z_i$ to obtain the following equation.

$$U_{i+1}\left(Z_i^T Z_i\right) = U_i\left(Z_i^T Z_i\right) + \Delta t \mathscr{P}_{U_i}^{\perp} \overline{\mathscr{L}} Z_i \tag{4.3}$$

To reiterate, the left-hand sides of equations (4.1) and (4.3) span the same space. So, our robust first-order retraction is obtained by just orthonormalizing (4.3) and then computing Z_{i+1} according to (4.2).

$$U_{i+1}^{\text{robust}} = \texttt{orth}\left(U_i\left(Z_i^T Z_i\right) + \Delta t \mathscr{P}_{U_i}^{\perp} \overline{\mathscr{L}} Z_i\right) \tag{4.4}$$

The orthonormalization may be done by a QR decomposition, SVD, or other algorithm of choice such as the re-orthonormalization procedure in [47].

Now we have a robust, first-order perturbative retraction. It would be nice to obtain robust higher-order corrections for Eqs. (2.6b, 2.6c, 2.6d). However, the above methodology does not extend to these higher-order corrections. This is because we need to sum the corrections in (2.5), which is not invariant to individual rotations of $\dot{\mathcal{U}}_j$. Fortunately, we can utilize our gradient-descent iterations (Alg. 3.1) with the robust first-order retraction (Eqs. 4.2 and 4.4) that utilizes a re-orthonormalization step in each inner gradient-descent iteration to obtain arbitrarily high-order retractions robust to overapproximation provided $X_i + \Delta t \overline{\mathscr{L}} \in \mathscr{M}_r$ or $\mathscr{P}_{\mathscr{M}_r}^{\perp}(X_i + \Delta t \overline{\mathscr{L}})$ decays sufficiently quickly (see section 3.2). Note, however, that high-order convergence of the projection-retraction error is not guaranteed, and a counterexample when the normal closure error $\varepsilon_{\mathcal{N}}$ dominates is provided in figure 5.

The robust retraction defined by Eqs. (4.2) and (4.4) avoids the error accumulation from the pseudoinverse. But, we pay for it by sacrificing mode continuity. In (4.4), we are integrating the span of U rather than individual modes, and so we do not keep track of how each mode evolves. In summary, there is a tradeoff, and the user can decide what is important for a particular problem. The pseudoinverse results in additional error accumulation but preserves mode continuity. The new robust retraction with gradient descent does not accumulate extra error, but it loses mode continuity. Both techniques preserve high-order convergence when the system is well-conditioned. However, to preserve high-order convergence when the system is ill-conditioned, time steps must be sufficiently small such that the distance from our initial point X_i and the fixed point $\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t \overline{\mathscr{L}})$ is smaller than the smallest singular value of X_i [34]. Geometrically, the low-rank manifold curvature tends to infinity, and these high-order corrections cannot capture areas of the manifold that are not smooth unless sufficiently small steps are taken.

4.1.2. Rank augmentation. As mentioned in Section 1, different algorithms and criteria have been developed to determine when to augment the rank of our system [17, 20, 46, 63]. Our approach, depicted in Figure 4, is to measure the angle between the full-rank direction $\overline{\mathscr{L}}$ and its projection onto the tangent space. This provides a non-dimensional parameter related to how fast the dynamics departs from the low-rank manifold. If the angle is larger than some threshold θ^*, we increment the rank by a pre-specified value r_{inc} up to a maximum rank of $r_{\max}$. To reduce rapid variations in the rank as a function of time, one can alter the rank of the solution only if not altered in recent time steps. Simple dynamical systems can be used to do this using a memory and forgetting rate, e.g., [37, 39, 54].

The next question becomes: how do we actually augment the rank of the system? First, we artificially augment $X_i = U_i Z_i^T$ as follows.

$$X_i = \begin{bmatrix} U_i & Q \end{bmatrix} \begin{bmatrix} Z_i^T \\ 0 \end{bmatrix} \equiv \hat{U}_i \hat{Z}_i^T \tag{4.5}$$

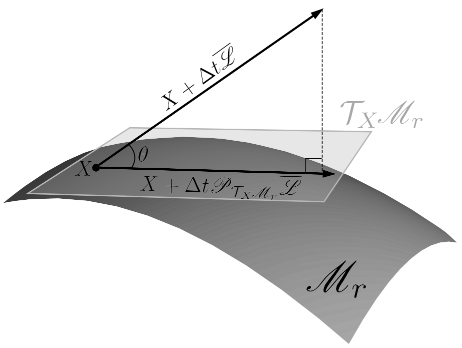
$X + \Delta t\overline{\mathscr{L}}$
$\mathcal{T}_X\mathscr{M}_r$
θ
X
$X + \Delta t\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}$
$\mathscr{M}_r$

Fig. 4: Depicted above is a point X on the low-rank manifold $\mathscr{M}_r$. From that point, the dynamics dictate we travel in direction $\overline{\mathscr{L}}$; this direction is not known a priori and changes within time integration steps, which our DORK schemes take into account. To stay on the manifold, the dynamics (to first-order) must be projected onto the tangent space. The angle θ between $\overline{\mathscr{L}}$ and $\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}$ is used as our metric to determine when to increase the rank of the solution.

Above, Q is chosen to be semi-orthonormal with columns that are orthogonal to the columns of U_i. Then, we may simply apply our retractions using $\hat{U}_i$ and $\hat{Z}_i$. However, we note that if we apply our update to U_{i+1} first, we will not update the subspace spanned by Q in $\hat{U}$ due to the zeros in $\hat{Z}$. As such, we recommend first updating $\hat{Z}$ via a simplified form of (4.2) below.

$$\hat{Z}_i \leftarrow \hat{Z}_i + \Delta t\overline{\mathscr{L}}^T\hat{U}_i \tag{4.6}$$

To compensate for the change in $\hat{Z}_i$, we must calculate $\overline{\mathscr{L}}'$ such that $U_iZ_i^T + \Delta t\overline{\mathscr{L}} = \hat{U}_i\hat{Z}_i^T + \Delta t\overline{\mathscr{L}}'$. Finally, we apply a retraction of our choice starting from point $\hat{U}_i\hat{Z}_i^T$ in direction $\overline{\mathscr{L}}'$ to update the span of $\hat{U}$ (in addition to updating $\hat{Z}$ again).

The ambiguity here is how to choose Q, i.e. how to augment our subspace U. In [46], Lin considers taking the first few left singular vectors of the residual of $\overline{\mathscr{L}}$, defined as $(I - \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r})\overline{\mathscr{L}}$, with the largest singular values [46]. This approach, while principled, can be expensive as it can require computing the SVD of $(I - \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r})\overline{\mathscr{L}}$ at each step. A computationally efficient implementation of [46]'s approach is to use the randomized SVD. Assuming $\overline{\mathscr{L}}$ is low-rank, obtaining the first few singular vectors of $(I - \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r})\overline{\mathscr{L}}$ is computationally cheap and provably accurate (see [27]). The randomized SVD inherently introduces stochastic variability into the rank augmentation algorithm, which is analyzed in [28]. Here, the randomized SVD is only used to choose the basis with which we augment the subspace, so it does not immediately induce a new error in integrating the dynamical system along the low-rank manifold. However, the stochastic variability may result in a slightly suboptimal choice of the basis Q. See Algorithm 4.1 for an implementation. We note that any retraction robust to overapproximation may be used in Algorithm 4.1, e.g. the robust perturbative retraction with automatic gradient descent, and it should ensure that U_{i+1} is orthonormalized.

4.1.3. Rank reduction. To reduce the rank of a point after integration, the process is straightforward [40, 16]. Because U_{i+1} is orthonormal, the singular values of X_{i+1} are defined by the eigenvalues of $Z_{i+1}^TZ_{i+1}$. As such, the impact of truncating the rank of X_{i+1} measured by the Frobenius norm is equal to the sum of the eigenvalues that we omit. Hence, we drop the smallest eigenvalues and their associated eigenvectors so long as the truncation results in a change in Frobenius norm less than a predetermined value σ^*. To remove the respective singular vectors from X_i, we project U_{i+1} and Z_{i+1} onto a truncated set of eigenvectors of $Z_{i+1}^TZ_{i+1}$. We refer to Algorithm 4.1 for details.

4.2. Illustration. To illustrate our robust retraction with gradient descent and rank augmentation, we consider the problem of rank discovery. Suppose we have a low-rank approx-

Algorithm 4.1 Angular rank-adaptive retraction with randomized SVD

```
Input: $X_i \in \mathscr{M}_{r_i}$, $\overline{\mathscr{L}} \in \mathbb{R}^{m\times n}$, $\Delta t \in \mathbb{R}$, $\theta^* \in [0,\pi/2]$, $\sigma^* \in [0,1]$, $r_{\text{inc}} \in \mathbb{Z}^+$, $r_{\max} \in \mathbb{Z}^+$
Output: $X_{i+1} \in \mathscr{M}_{r_{i+1}}$
$r_i = \operatorname{rank}(X_i)$
$\dot{U}_i = \mathscr{P}^{\perp}_{U_i}\overline{\mathscr{L}} Z_i (Z_i^T Z_i)^{-1}, \quad \dot{Z}_i = \overline{\mathscr{L}}^T U_i$
$\mathscr{P}_{T_X\mathscr{M}_r}\overline{\mathscr{L}} = U_i\dot{Z}_i + \dot{U}_i Z_i^T$
$\theta = \cos^{-1}\left(\frac{\|\mathscr{P}_{T_X\mathscr{M}_r}\overline{\mathscr{L}}\|}{\|\overline{\mathscr{L}}\|}\right)$
if $((\theta^* = 0)$ or $(\theta > \theta^*))$ and $(r_i < r_{\max})$ then
    $r_{\text{inc}} = \min(r_i, r_{\text{inc}}, r_{\max} - r_i)$
    $Q = \texttt{randSVD}\left(\mathscr{P}^{\perp}_{U_i}\overline{\mathscr{L}}\right)$ via [27, alg. 4.4]
    $\hat{U}_i = \begin{bmatrix} U_i & Q \end{bmatrix}, \quad Z_i = \begin{bmatrix} Z_i & \texttt{zeros}(n, r_{\text{inc}}) \end{bmatrix}$
    $\hat{Z}_i \leftarrow \hat{Z}_i + \Delta t \overline{\mathscr{L}}^T \hat{U}_i$
    $\overline{\mathscr{L}} \leftarrow \overline{\mathscr{L}} - (\hat{U}_i \hat{Z}_i^T - U_i Z_i^T)/\Delta t$
end if
$X_{i+1} = \mathcal{R}_{X_i}(\Delta t \overline{\mathscr{L}} - (X_{i+1} - X_i))$
$V, \Lambda = \texttt{eig}\left(Z_i^T Z_i\right)$ with eigenvalues (and associated eigenvectors) sorted in ascending
    order
cutoff $=$ last index $j$ where $(\sum_{i=1}^{j} \Lambda(i)/\sum_i \Lambda(i))^{1/2} < \sigma^*$
$V \leftarrow V(:, \text{cutoff}+1 : \text{end})$
$U_{i+1} \leftarrow U_{i+1}V, \quad Z_{i+1} \leftarrow Z_{i+1}V$
```

imation X to $\mathfrak{X}$, where $X \in \mathbb{R}^{500\times 220}$ has rank 20, and $\mathfrak{X} = X + \Delta t \overline{\mathscr{L}}$ has rank 125, where X and $\overline{\mathscr{L}}$ have Frobenius norm one, and $\Delta t = 0.1$. The matrices X and $\overline{\mathscr{L}}$ are initialized randomly as in Sect. 2.2.

Our goal is to recover the full-rank $\mathfrak{X}$ efficiently using our retractions. We combine our rank-adaptive retraction 4.1 and our automatic gradient-descent retraction 3.2 to create an iterative scheme that keeps increasing the rank of X until the normalized local retraction error ε_l has norm below a specified threshold, ε_l^*. We specify the details of the implementation in Algorithm 4.2, where we set the incremental rank increase to 25, the max rank to 200, $\varepsilon_l^* = 10^{-6}$, and the maximum number of iterations in gradient descent $N_{\max} = 16$.

Algorithm 4.2 Automatic, adaptive retraction for rank discovery and compression

```
Input: $X_i \in \mathscr{M}_{r_i}$, $\overline{\mathscr{L}} = U_L Z_L^T \in \mathbb{R}^{m\times n}$, $\Delta t \in \mathbb{R}$, $N \in \mathbb{Z}^+$, $\Delta^* \in \mathbb{R}$, $r_{\text{inc}} \in \mathbb{Z}^+$, $r_{\max} \in \mathbb{Z}^+$,
    $\varepsilon_l^* \in \mathbb{R}$, $N_{\max} \in \mathbb{Z}^+$
Output: $X_{i+1} \in \mathscr{M}_{r_{i+1}}$
$X_{i+1} = X_i$
while $\|X_i + \Delta t \overline{\mathscr{L}} - X_{i+1}\|/\|X_i\| > \varepsilon_l^*$ do
    $X_{i+1} = \texttt{rankAdaptiveRetraction}(X_{i+1}, \overline{\mathscr{L}} - (X_{i+1} - X_i)/\Delta t = U_L Z_L^T, \Delta t, \theta^* =$
    $0, \sigma^* = \varepsilon_l^*, r_{\text{inc}}, r_{\max}, N_{\max})$ using automatic gradient-descent retraction internally with
    $\Delta^* = \varepsilon_l^*$
end while
```

Figure 5 shows different errors from Table 2 as a function of iteration number as well as how the rank changes. When we underestimate the rank of the solution in the first few outer steps of the algorithm, the local retraction error and the normal closure error stay relatively

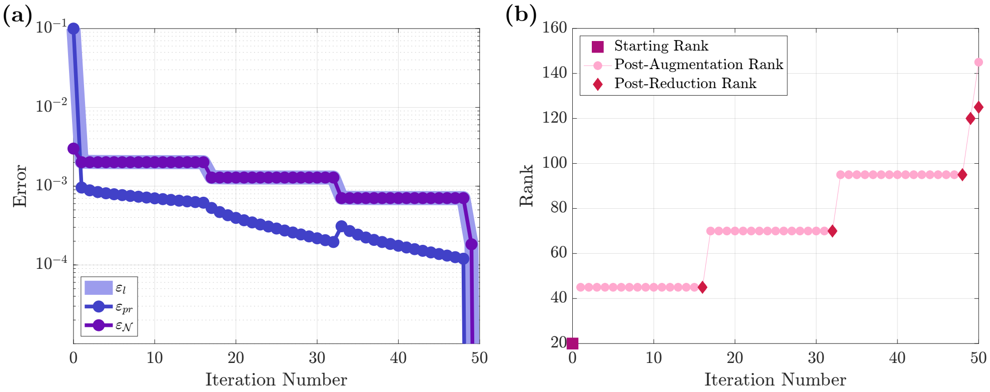

Fig. 5: Rank discovery using algorithm 4.2 with $\varepsilon_l^* = 10^{-6}$. (a) Local retraction error, projection-retraction error, and normal closure error (see Table 2). During the internal iterations of the gradient-descent retraction, the projection-retraction decreases. Then, when the rank is increased, the projection-retraction error increases since we are now measuring the error between our current approximation and new, higher-rank matrix. The normal closure error remains constant during the inner gradient-descent iterations since it is not a function of the retractions. Of course, it monotonically decreases as we increase the rank. The local retraction error remains almost constant during the inner iterations of gradient descent since it includes $\varepsilon_{\mathcal{N}}$, which dominates the total error. (b) Rank of each iteration as a result of augmentation as well as truncation. Note that the truncation only takes effect once we exceed the true rank of the solution at the last iteration.

large during the inner gradient-descent steps since they include error in the normal space. We see the projection-retraction error decrease, albeit quite slowly, when we underapproximate the rank. This is in contrast to what we observed in section 3, where we rapdily converged to the best low-rank approximation. The difference here is that we are starting without a good initial guess for the subspace spanned by U after augmenting the rank. Essentially, we are trying to search a high-dimensional vector space via perturbations, so convergence is slow. What's more, in optimizing the span of U, there exist local minima for the left singular vectors of $\mathfrak{X}$ with associated singular values that are not the largest in magnitude. For example if we set U to the left singular vectors with the smallest singular values, we would still reach a stationary point where $\mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \overline{\mathscr{L}} = 0$. The saving grace of this algorithm, though, is that when we hit or exceed the rank of the full-rank solution, all of the errors plumet due to the quadratic convergence of the projection-retraction error and the fact that $\varepsilon_{\mathcal{N}} = 0$.

5. Novel DORK schemes.

In this Section, we build off our optimal perturbative and gradient-descent retractions to develop two new low-rank integration schemes based off the Dynamically Orthogonal Runge-Kutta (DORK) schemes [12]. These schemes integrate along the low-rank manifold, updating the subspace upon which we project the system's dynamics in stages during the time step. To do so, instead of taking $\overline{\mathscr{L}}$ to be a constant $\mathcal{O}(1)$ term, we expand it as a perturbation series itself.

$$\Delta t \overline{\mathscr{L}} \equiv \sum_{j=1}^{\infty} \Delta t^j \overline{\mathcal{L}}_j \tag{5.1}$$

We denote the partial sum $\overline{\mathscr{L}}^{(k)} = \sum_{j=1}^{k} \Delta t^j \overline{\mathcal{L}}_j$, which is a kth-order integration scheme, and each $\overline{\mathcal{L}}_j$ is a jth-order correction to $\overline{\mathscr{L}}^{(j-1)}$ such that $\overline{\mathscr{L}}^{(j)}$ increases an order of accuracy. To construct integration schemes as a perturbation series, we refer to [12].

5.1. so-DORK schemes.

First, we derive the stable, optimal Dynamically Orthogonal Runge-Kutta (so-DORK) schemes. The process is straightforward; we substitute (2.5) and (5.1) into (2.4) and solve for the first- through fourth-order corrections.

$$\dot{\mathcal{U}}_1 = \mathscr{P}^{\perp}_{U_i} \overline{\mathcal{L}}_1 Z_i (Z_i^T Z_i)^{-1} \tag{5.2a}$$

$$\dot{\mathcal{U}}_2 = \left[\mathscr{P}^{\perp}_{U_i}\left(\overline{\mathcal{L}}_1\overline{\mathcal{L}}_1^T U_i + \overline{\mathcal{L}}_2 Z_i\right) - \dot{\mathcal{U}}_1\left(U_i^T\overline{\mathcal{L}}_1 Z_i + Z_i^T\overline{\mathcal{L}}_1^T U_i\right)\right](Z_i^T Z_i)^{-1} \tag{5.2b}$$

$$\begin{aligned}\dot{\mathcal{U}}_3 = \Big[&\mathscr{P}^{\perp}_{U_i}\left(\overline{\mathcal{L}}_1\overline{\mathcal{L}}_1^T\dot{\mathcal{U}}_1 + \left(\overline{\mathcal{L}}_1\overline{\mathcal{L}}_2^T + \overline{\mathcal{L}}_2\overline{\mathcal{L}}_1^T\right)U_i + \overline{\mathcal{L}}_3 Z_i\right)\\ &- \dot{\mathcal{U}}_2\left(U_i^T\overline{\mathcal{L}}_1 Z_i + Z_i^T\overline{\mathcal{L}}_1^T U_i\right) - \dot{\mathcal{U}}_1\Big(U_i^T\overline{\mathcal{L}}_2 Z_i + Z_i^T\overline{\mathcal{L}}_2^T U_i\\ &+ U_i^T\overline{\mathcal{L}}_1\overline{\mathcal{L}}_1^T U_i + \dot{\mathcal{U}}_1^T\overline{\mathcal{L}}_1 Z_i + Z_i^T\overline{\mathcal{L}}_1^T\dot{\mathcal{U}}_1 - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1 Z_i^T Z_i\Big)\Big](Z_i^T Z_i)^{-1}\end{aligned} \tag{5.2c}$$

$$\begin{aligned}\dot{\mathcal{U}}_4 = \Big[&\mathscr{P}^{\perp}_{U_i}\Big(\overline{\mathcal{L}}_1\overline{\mathcal{L}}_1^T\dot{\mathcal{U}}_2 + \left(\overline{\mathcal{L}}_1\overline{\mathcal{L}}_2^T + \overline{\mathcal{L}}_2\overline{\mathcal{L}}_1^T\right)\dot{\mathcal{U}}_1 + \left(\overline{\mathcal{L}}_1\overline{\mathcal{L}}_3^T + \overline{\mathcal{L}}_3\overline{\mathcal{L}}_1^T + \overline{\mathcal{L}}_2\overline{\mathcal{L}}_2^T\right)U_i\\ &+ \overline{\mathcal{L}}_4 Z_i) - \dot{\mathcal{U}}_3\left(U_i^T\overline{\mathcal{L}}_1 Z_i + Z_i^T\overline{\mathcal{L}}_1^T U_i\right) - \dot{\mathcal{U}}_2\Big(U_i^T\overline{\mathcal{L}}_2 Z_i + Z_i^T\overline{\mathcal{L}}_2^T U_i\\ &+ U_i^T\overline{\mathcal{L}}_1\overline{\mathcal{L}}_1^T U_i + \dot{\mathcal{U}}_1^T\overline{\mathcal{L}}_1 Z_i + Z_i^T\overline{\mathcal{L}}_1^T\dot{\mathcal{U}}_1 - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1 Z_i^T Z_i\Big) - \dot{\mathcal{U}}_1\,(U_i^T\overline{\mathcal{L}}_3 Z_i\\ &+ Z_i^T\overline{\mathcal{L}}_3^T U_i + U_i^T\left(\overline{\mathcal{L}}_2\overline{\mathcal{L}}_1^T + \overline{\mathcal{L}}_1\overline{\mathcal{L}}_2^T\right)U_i + \dot{\mathcal{U}}_1^T\overline{\mathcal{L}}_2 Z_i + Z_i^T\overline{\mathcal{L}}_2^T\dot{\mathcal{U}}_1\\ &+ U_i^T\overline{\mathcal{L}}_1\overline{\mathcal{L}}_1^T\dot{\mathcal{U}}_1 + \dot{\mathcal{U}}_1\overline{\mathcal{L}}_1\overline{\mathcal{L}}_1^T U_i + \dot{\mathcal{U}}_2^T\overline{\mathcal{L}}_1 Z_i + Z_i^T\overline{\mathcal{L}}_1^T\dot{\mathcal{U}}_2 - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_2 Z_i^T Z_i\\ &- \dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_1 Z_i^T Z_i - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1 U_i^T\overline{\mathcal{L}}_1 Z_i - \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1 Z_i^T\overline{\mathcal{L}}_1^T U_i\Big)\Big](Z_i^T Z_i)^{-1}\end{aligned} \tag{5.2d}$$

We first remark that the additional high-order $\overline{\mathcal{L}}_j$ are not in equations (2.6a-2.6d) as they then remain within $\overline{\mathscr{L}}$. Second, the update equation for the coefficients Z_{i+1} is (4.2) or (2.2) which leads to stability properties. The following lemma follows directly from lemma 2.1.

LEMMA 5.1. *Starting from X_i, let X_{i+1} be the result of integrating $\Delta t\overline{\mathscr{L}}$ with an so-DORK scheme given by equations (5.2a-5.2d) and (2.2). Then $\|X_{i+1}\| \leq \|X_i + \Delta t\overline{\mathscr{L}}\|$.*

Proof. Let $X_{i+1} = U_{i+1}Z_{i+1}^T$. By construction, $U_{i+1} \in \mathcal{V}_{m,r}$, so $\|X_{i+1}\| = \|Z_{i+1}\|$. By (2.2), $\|Z_{i+1}\| = \|(X_i + \Delta t\overline{\mathscr{L}})^T U_{i+1}\| \leq \|X_i + \Delta t\overline{\mathscr{L}}\|$. □

This implies that if the Runge-Kutta integration scheme of $\mathscr{L}(X,t;\omega)$ is stable, its so-DORK counterpart will also be stable, making so-DORK special. Note that full-rank stability integrating $\mathscr{L}(\mathfrak{X},t;\omega)$ does not guarantee low-rank stability due to the dynamical model closure error ε_D and normal closure error $\varepsilon_{\mathcal{N}}$ (see table 2). However, in practice, these errors are small if the DLRA is expected to perform well; as a result, they tend not to affect stability.

5.2. gd-DORK schemes. Now, we utilize our gradient-descent retractions from Section 3 and develop gradient-descent Dynamically Orthogonal Runge-Kutta (gd-DORK) schemes. This family of integration schemes also updates the subspace as we integrate, but instead of approximating the high-order curvature of the low-rank manifold from one fixed starting point, the manifold is locally approximated at points along our trajectory. A kth-order integration scheme is given in Algorithm 5.1; note that we incorporate perturbations to $\overline{\mathscr{L}}$ as we integrate. For computational efficiency, we typically use a first-order retraction in each substep, though any order retraction may be used. Furthermore, an automatic variant may also be derived from Algorithm 3.2. Finally, we recall that even for gd-DORK, X_{i+1} is not formed; instead, U_{i+1} and Z_{i+1}^T are updated during the gradient descent inner iterations, as explained at the end of section 3.1.

LEMMA 5.2. *Given a close enough starting point and $\|X_i\| \sim \|\overline{\mathscr{L}}^{(j)}\|$ for all j, gd-DORK schemes are consistent, kth-order accurate integration schemes.*

Proof. We will show that, starting from a point $X_i \in \mathscr{M}_r$, Algorithm 5.1 asymptotically approximates $\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}^{(k)})$. We will proceed inductively, starting with one initial retraction.

Algorithm 5.1 gd-DORK Schemes

```
Input: $X_i \in \mathscr{M}_r$, $\{\overline{\mathscr{L}}^{(j)} \in \mathbb{R}^{m\times n}\}_{j=1}^{k}$, $\Delta t \in \mathbb{R}$
Output: $X_{i+1} \in \mathscr{M}_r$
$X_{i+1} = X_i$
for $j = 1 : k$ do
    $X_{i+1} \leftarrow \mathcal{R}_{X_{i+1}}(\Delta t\overline{\mathscr{L}}^{(j)} - (X_{i+1} - X_i))$
end for
```

Let $X^{(1)} = \mathcal{R}^{(p)}_{X_i}\left(\Delta t\overline{\mathscr{L}}^{(1)}\right)$, where $\mathcal{R}^{(p)}$ is a retraction that asymptotically approximates $\mathscr{P}_{\mathscr{M}_r}$ to the pth order, and $p \geq 2$. Assuming that $\|X_i\| \sim \|\overline{\mathscr{L}}^{(j)}\|$ for all j, we have:

$$X^{(1)} = \mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(1)}\right) + \mathcal{O}(\Delta t^p).$$

On iteration j, we will assume that we have $X^{(j-1)} = \mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}^{(j-1)}) + \mathcal{O}(\Delta t^q)$, where $q \geq j$. Certainly for $j = 2$, this is true. We will analyze the error of another step of gradient descent of sec. 3, using (3.1), (3.2), and the assumption about $X^{(j-1)}$.

$$\begin{aligned}
X^{(j)} &= \mathcal{R}^{(p)}_{X^{(j-1)}}\left(\Delta t\overline{\mathscr{L}}^{(j)} + (X_i - X^{(j-1)})\right)\\
&= \mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(j)}\right) + \mathcal{O}\left(\left\|\Delta t\overline{\mathscr{L}}^{(j)} + (X_i - X^{(j-1)})\right\|^p\right)\\
&= \mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(j)}\right)\\
&\quad + \mathcal{O}\left(\left\|\Delta t\overline{\mathscr{L}}^{(j)} + X_i - \mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}^{(j-1)}) + \mathcal{O}\left(\Delta t^q\right)\right\|^p\right)
\end{aligned} \tag{5.3}$$

Next, we decompose the full-rank point we are approximating, $X_i + \Delta t\overline{\mathscr{L}}^{(j)}$, into two components, and we then use the fact that $\overline{\mathscr{L}}^{(j)}$ is a $\mathcal{O}(\Delta t^j)$ correction to $\overline{\mathscr{L}}^{(j-1)}$.

$$\begin{aligned}
X_i + \Delta t\overline{\mathscr{L}}^{(j)} &= \mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(j)}\right) + \mathscr{P}^{\perp}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(j)}\right)\\
&= \mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(j-1)}\right) + \mathcal{O}\left(\Delta t^j\right) + \mathscr{P}^{\perp}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(j)}\right)
\end{aligned} \tag{5.4}$$

We now substitute (5.4) into (5.3).

$$X^{(j)} = \mathscr{P}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(j)}\right) + \mathcal{O}\left(\left\|\mathcal{O}\left(\Delta t^j, \Delta t^q\right) + \mathscr{P}^{\perp}_{\mathscr{M}_r}\left(X_i + \Delta t\overline{\mathscr{L}}^{(j)}\right)\right\|^p\right) \tag{5.5}$$

Letting $\varepsilon^{(j)}_{pr}$ denote the projection-retraction error at iteration j and recalling that $q \geq j$, if $X_i + \Delta t\overline{\mathscr{L}}^{(j)} \in \mathscr{M}_r$, we have that $\|\varepsilon^{(j)}_{pr}\| \sim \Delta t^{jp}$. Otherwise, we refer to the proof in [34] that shows the orthogonal error decays provided that we start from a point with a distance to the fixed point $\mathscr{P}_{\mathscr{M}_r}(X_i + \Delta t\overline{\mathscr{L}}^{(j)})$ less than the smallest singular value of the fixed point. Hence by induction, we have shown that the gd-DORK scheme is consistent and preserves the order of convergence of the full-rank integration scheme. □

Analogously to the gradient-descent retractions, if we use stable, optimal retractions as our inner retraction, the resulting gd-DORK scheme will be stable, following directly from lemma 3.1.

LEMMA 5.3. *Starting from X_i, let X_{i+1} be the result of integrating $\Delta t\overline{\mathscr{L}}$ with a gd-DORK scheme given by algorithm 5.1 using an optimal retraction $\mathcal{R}^{opt}$ given by equations (2.6a-2.6d) and (2.2) as the final inner retraction. Then $\|X_{i+1}\| \leq \|X_i + \Delta t\overline{\mathscr{L}}\|$.*

Proof. Assume we perform k inner iterations of gradient descent. From the last iteration of the gradient-descent retraction, we have that

$$\begin{aligned}\left\|X_{i+1}^{(k)}\right\| &= \left\|\mathcal{R}^{\mathrm{opt}}_{X_{i+1}^{(k-1)}}\left(X_i + \Delta t\overline{\mathscr{L}}^{(k)} - X_{i+1}^{(k-1)}\right)\right\| \\ &\leq \|X_{i+1}^{(k-1)} + X_i + \Delta t\overline{\mathscr{L}}^{(k)} - X_{i+1}^{(k)}\| \\ &= \|X_i + \Delta t\overline{\mathscr{L}}^{(k-1)}\|\end{aligned}$$

The inequality above follows from lemma 2.1, and recall that $\overline{\mathscr{L}} = \overline{\mathscr{L}}^{(k)}$ for a kth-order accurate integration scheme. □

As a final remark, we note a key difference between so-DORK and gd-DORK that affects the algorithmic implementation and is linked to (5.1). The so-DORK schemes operate on updates $\overline{\mathcal{L}}_j$ to the differential operator $\overline{\mathscr{L}}^{(j-1)}$ while gd-DORK schemes operate on the jth-order integrator $\overline{\mathscr{L}}^{(j)}$ itself. As such, unless updates to $\overline{\mathscr{L}}^{(j)}$ can be computed along the way (as in Heun's method), the most accurate $\overline{\mathscr{L}}^{(j)}$ available at each gradient-descent step should be used. That is, for higher-order Runge-Kutta methods where points and functions must be evaluated outside of the integration path, it is more accurate to pre-compute a high-order approximation to $\overline{\mathscr{L}}$ rather than increment the integration order one-by-one as in Alg. 5.1.

6. Numerical experiments.

6.1. Matrix differential equation.

First, we compare the efficacy of our new DORK schemes to other integration schemes in the literature on a system of linear oscillators. As in [12, sec. 7.2], we solve for a rank-16 approximation of a 26×26 system governed by $\ddot{\mathfrak{X}} = -\Omega^2\mathfrak{X}$ with initial conditions $\mathfrak{X}(0) = \mathfrak{X}_0$, $\dot{\mathfrak{X}}(0) = \dot{\mathfrak{X}}_0$). We select $\Omega = \mathrm{diag}(\omega_1, \omega_1, \omega_2, \omega_2, \ldots, \omega_{13}, \omega_{13})$ with each ω_i independently chosen from a standard normal distribution. We construct a matrix $R(t) \in \mathbb{R}^{26\times 26}$ as a tridiagonal filled with 2×2 block rotation matrices $R_1, \ldots, R_{13}$ such that,

$$R_i(t) = \begin{bmatrix}\cos(\omega_i t) & -\sin(\omega_i t)\\ \sin(\omega_i t) & \cos(\omega_i t)\end{bmatrix},$$

and then $R(t) = \mathrm{diag}(R_1, \ldots, R_{13})$. In contrast to the example in [12], $S \in \mathbb{R}^{26\times 26}$ is a diagonal matrix with non-increasing entries such that the singular values of the system are $S_{ii} = 100 + 10z_i$ for $1 \leq i \leq 14$ and $S_{ii} = 10^{-5(1+(i-14)/12)}$ for $15 \leq i \leq 26$, where z_i are realizations of independent, standard normal random variables. By construction, our system is ill-conditioned with two small singular values out of the sixteen that are not truncated; the condition number of the rank-16 truncated version of $S^T S$ in the Frobenius norm is approximately $3.00 \cdot 10^{15}$. We construct a new matrix $Q \in \mathbb{R}^{26\times 26}$ by orthonormalizing a matrix with independent samples from a uniform distribution. It is simple to verify that the exact solution is given by $\mathfrak{X}(t) = R(t)QS$ with $\mathfrak{X}_0 = R(0)QS$ and $\dot{\mathfrak{X}}_0 = \dot{R}(0)QS$. To apply our DORK schemes, we integrate an augmented state $\begin{bmatrix}X & \dot{X}\end{bmatrix}$.

Figure 6 shows the error measured in the Frobenius norm, normalized by the norm of the initial conditions. Compared to the original DORK schemes, so-DORK and gd-DORK are more accurate; they reduce the rate at which error is accumulated, and they preserve high-order convergence that is eliminated by the ill-conditioned matrix inversion in the original DORK schemes. We note however that so-DORK does not converge all the way to the best approximation. This is due to error accumulated by the pseudoinverse. In selecting the threshold at which we truncate singular values before inverting our correlation matrix, we must balance the error incurred by inverting a singular matrix and the error incurred by approximating our correlation matrix. The gd-DORK scheme avoids this by using the robust

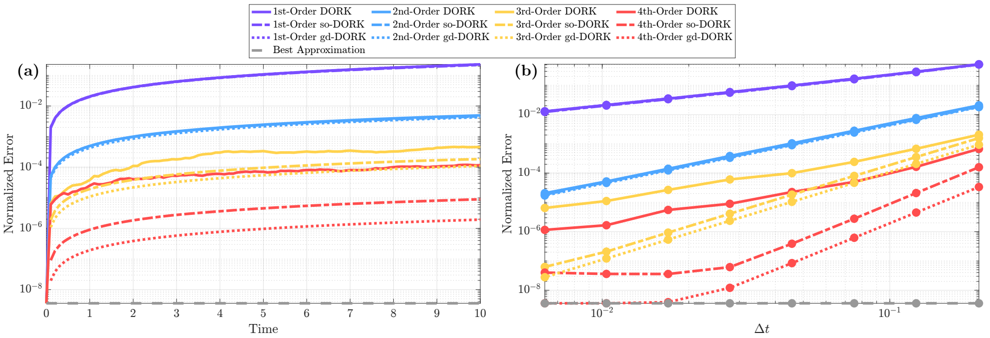

Fig. 6: Matrix differential equation. (a) Error accumulation vs. time for twelve numerical integration schemes for $t \in [0,10]$ and $\Delta t = 0.1$. As a threshold for truncating singular values for pseudoinversion, we use $10^{-9}\|X(t)\|$. The so-DORK and gd-DORK methods outperform classic DORK, especially at high order. (b) Convergence of the error at $t = 10$ for multiple values of Δt. The so-DORK and gd-DORK schemes preserve high-order convergence even when the system is ill-conditioned.

N_t	PRK	Proj.-Split.	DORK	so-DORK	gd-DORK
50	$2.11 \cdot 10^{-2}$	$2.11 \cdot 10^{-2}$	$2.06 \cdot 10^{-2}$	$1.86 \cdot 10^{-2}$	$1.80 \cdot 10^{-2}$
134	$2.86 \cdot 10^{-3}$	$2.86 \cdot 10^{-3}$	$2.80 \cdot 10^{-3}$	$2.63 \cdot 10^{-3}$	$2.43 \cdot 10^{-3}$
968	$5.40 \cdot 10^{-5}$	$5.40 \cdot 10^{-5}$	$5.20 \cdot 10^{-5}$	$4.99 \cdot 10^{-5}$	$4.62 \cdot 10^{-5}$

Table 3 Matrix differential equation. Frobenius norms of the final errors after integrating for $t \in [0,10]$ with the projected Runge-Kutta (PRK), projector-splitting, DORK, so-DORK, and gd-DORK second-order integrators, for three values of the number of time steps, N_t. DORK schemes are more accurate by incorporating dynamic subspace updates as we integrate, and so-DORK and gd-DORK are most accurate.

scheme (4.4, 4.2) in each gradient descent step. This entirely avoid matrix inversion at the cost of losing mode continuity.

In addition, we compare the error accumulated by the three DORK schemes to state-of-the-art numerical schemes. Namely, we implemented the projected Runge-Kutta (PRK) schemes [32] and the symmetrized projector-splitting integrator [49]. All integration schemes are of second-order accuracy, and results are displayed in Table 3. Overall, we see that all of the DORK schemes improve the projector-splitting and projected Runge-Kutta methods as they update the subspace onto which we project the system dynamics between time steps. Furthermore, the so-DORK and gd-DORK methods outperform classic DORK.

6.2. Advection-diffusion partial differential equation. As our next example, we solve the 2D linear advection-diffusion PDE,

$$\frac{\partial u}{\partial t} + c \cdot \nabla u = \nu \nabla^2 u. \tag{6.1}$$

We impose periodic boundary conditions on a $[0,1]^2$ domain. This domain is discretized in space by 256 points in each dimension using a second-order central finite difference scheme. For time $t \in [0,5]$, to emphasize curvature-related errors, we employ $\Delta t = 10^{-3}$ with a variety of time-integration schemes. For our velocity field c, we consider potential flow with some background horizontal velocity superposed with a Rankine vortex.

$$c_x = \frac{4}{3} + \begin{cases} -\frac{\Gamma y_o}{R^2} & \sqrt{x_o^2 + y_o^2} < R \\ -\frac{\Gamma y_o}{x_o^2 + y_o^2} & \sqrt{x_o^2 + y_o^2} \geq R \end{cases}, \quad c_y = \begin{cases} \frac{\Gamma x_o}{R^2} & \sqrt{x_o^2 + y_o^2} < R \\ \frac{\Gamma x_o}{x_o^2 + y_o^2} & \sqrt{x_o^2 + y_o^2} \geq R \end{cases}$$

r	PRK	Proj.-Split.	DORK	so-DORK	gd-DORK
5	$2.20\cdot 10^{-1}$	$2.13\cdot 10^{-1}$	$2.07\cdot 10^{-1}$	$2.09\cdot 10^{-1}$	$2.14\cdot 10^{-1}$
15	$7.68\cdot 10^{-3}$	$7.61\cdot 10^{-3}$	DNC	$1.04\cdot 10^{-2}$	$7.53\cdot 10^{-3}$

Table 4 Normalized, time-averaged errors from integrating (6.1) with second-order accurate low-rank schemes in space and time. At rank-5, the DORK and so-DORK are most accurate possibly due to their high-order approximation of the low-rank manifold curvature. At rank-15, the system is ill-conditioned, so the classic DORK scheme does not converge. The so-DORK scheme incurs slightly more error due to the pseudoinversion, and the gd-DORK scheme is most accurate.

We choose $\Gamma = 8R/3$, $R = 1/16$, and offset variables $x_o = x - 0.5$, $y_o = y - 0.5$. Our initial conditions are haphazardly placed Gaussians,

$$u(x,y,t=0) = \sum_{i=1}^{7} a_i \exp\left(-\frac{(x-\mu_i^{(x)})^2 + (y-\mu_i^{(y)})^2}{2\sigma_i^2}\right), \tag{6.2}$$

where

$$\begin{aligned}
a &= \begin{bmatrix}1, & 1, & 1, & 2, & 1, & 1, & 1\end{bmatrix}^T \\
\mu^{(x)} &= \begin{bmatrix}0.5, & 0.3, & 0.354, & 0.4, & 0.15, & 0.65, & 0.35\end{bmatrix}^T \\
\mu^{(y)} &= \begin{bmatrix}0.5, & 0.5, & 0.5, & 0.35, & 0.65, & 0.25, & 0.65\end{bmatrix}^T \\
\sigma^2 &= \begin{bmatrix}200, & 500, & 400, & 400, & 400, & 450, & 450\end{bmatrix}^T.
\end{aligned}$$

To efficiently evolve the system, we truncate the rank of c_x and c_y to a rank-4 approximation, which is the threshold where singular values are less than 10^{-1} times the largest singular value, and we use these approximate velocity fields for both the full-rank and low-rank simulations so that we may directly compare the integration error between numerical schemes. In Figure 7, we show full-rank and low-rank solutions computed with Heun's method and the second-order so-DORK scheme, respectively. We fix the rank to $r = 5$ to contrast with our rank-adaptive solution which dynamically adjusts the rank to capture feature solutions on the fly. Clearly, the rank-adaptive solution is preferable to the under-resolved rank-5 solution. The rank initially increases to accommodate for the advected field hitting the vortex, but then the rank decreases as diffusion dominates the system. The adaptive-rank algorithm 4.1 tracks the rank of the full-order solution quite well. Adjusting σ^* and θ^* could yield an even closer match at an increased computational cost.

We also integrate the system with the same integrators from the literature as in Section 6.1. The mean error (as measured by the deviation from the full-rank numerical solution) divided by the norm of the initial conditions is enumerated in Table 4 at different ranks. At lower ranks, the DORK and so-DORK schemes perform the best, possibly because they directly incorporate the high-order curvature of the low-rank manifold, and the system is not ill-conditioned, meaning the perturbation series converge quickly. At higher ranks, the classic DORK scheme does not converge due to the singular correlation matrix. This is where the gd-DORK scheme excels since in each substep, it only projects onto the tangent space of the low-rank manifold while still updating the subspace as we integrate.

6.3. Stochastic Fisher-KPP PDE. In this section, we showcase the flexibility of our gradient-descent retractions by employing them in an implicit-explicit (IMEX) numerical integration scheme rather than an explicit Runge-Kutta scheme. We solve the stochastic Fisher-KPP equation, a reaction–diffusion PDE that, for example, describes biological population or chemical reaction dynamics with diffusion [19, 35].

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + r(\omega)u(1-u) \tag{6.3}$$

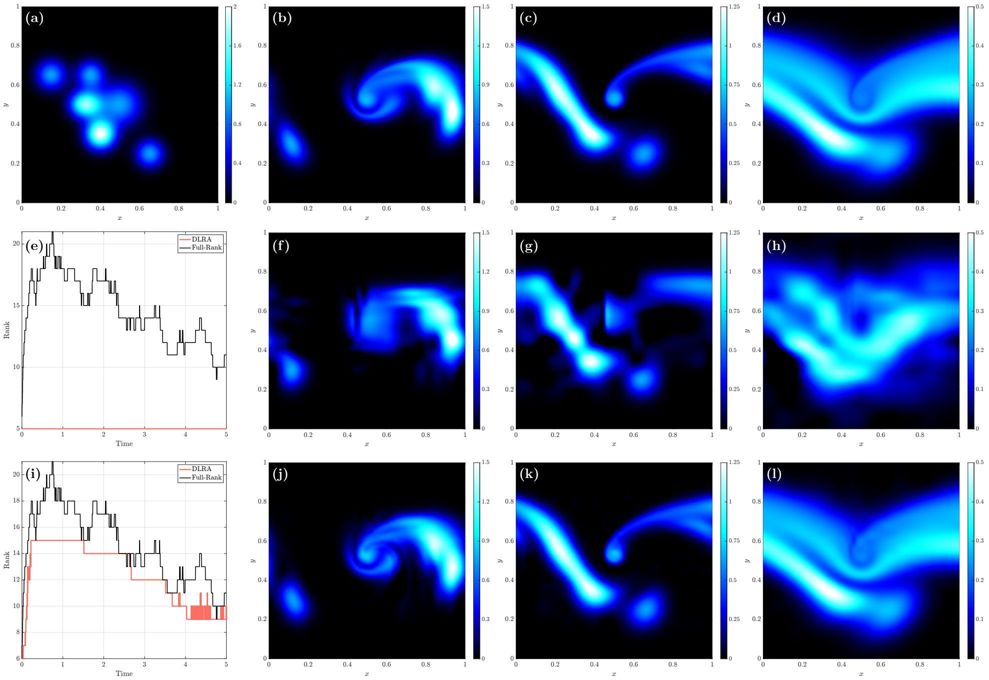

Fig. 7: Advection-diffusion PDE. (a) Initial conditions (6.2). (b-d) Full-rank solution at $t = 0.3$, 0.6, and 3.0, respectively. (e) Rank as a function of time: fixed rank-5 (red) and rank of the full-order simulation thresholded at $\sigma^* = 2 \cdot 10^{-3}$ (black). (f-h): Rank-5 solution at the aforementioned three times. (i) As (e), but for the *adaptive* rank as a function of time with $\theta^* = 0.1$, same $\sigma^* = 2 \cdot 10^{-3}$, $r_{\text{inc}} = 1$, and $r_{\max} = 20$ (red). (j-l) Corresponding rank-adaptive solution at the same three times. The rank-adaptive solution captures the full-rank features while eliminating redundant degrees of freedom.

The PDE (6.3) is stochastic (S-PDE) because we model the reaction rate r as a random variable $r \sim \mathcal{U}[1/4, 1/2]$, where ω denotes a simple event in the event space Ω. In general, (6.3) admits traveling waves as solutions and is used in ecology [45, 15], biology [57], and more [22]. It also forms a building block of more complex ocean biogeochemical models [2, 42, 23, 24].

Presently, we consider the domain $x \in [0, 40]$ and time duration $t \in [0, 12.5]$. For boundary conditions, we assume zero diffusive fluxes at the two boundaries,

$$\frac{\partial u}{\partial x}(0,t) = 0\ , \quad \frac{\partial u}{\partial x}(40,t) = 0.$$

The initial conditions are stochastic, of the following form.

$$u(x,0;\omega) = a(\omega)e^{-b(\omega)x^2}, \quad a \sim \mathcal{U}[1/5, 2/5], \quad b \sim \mathcal{U}[1/10, 11/10]$$

The random variables a, b, r are all independent, and we set the diffusion coefficient $D = 1$.

To solve the S-PDE numerically, we employ two approaches: Monte Carlo (MC) realizations and a DLRA of the MC realizations, where each column of our solution matrix at a fixed time represents one stochastic realization. We use an implicit-explicit Crank-Nicholson leapfrog time integration scheme [31], treating the diffusion term with Crank-Nicholson and the reaction term explicitly with leapfrog. Such an integration scheme is numerically efficient because we separate the deterministic and stochastic operators; by treating the stochastic operator explicitly, we avoid having to invert a different matrix for each column/realization of the Monte Carlo solution. To initialize the numerical integration, as the leapfrog scheme requires the solution at 2 previous time steps, we use a first-order time-integration scheme for

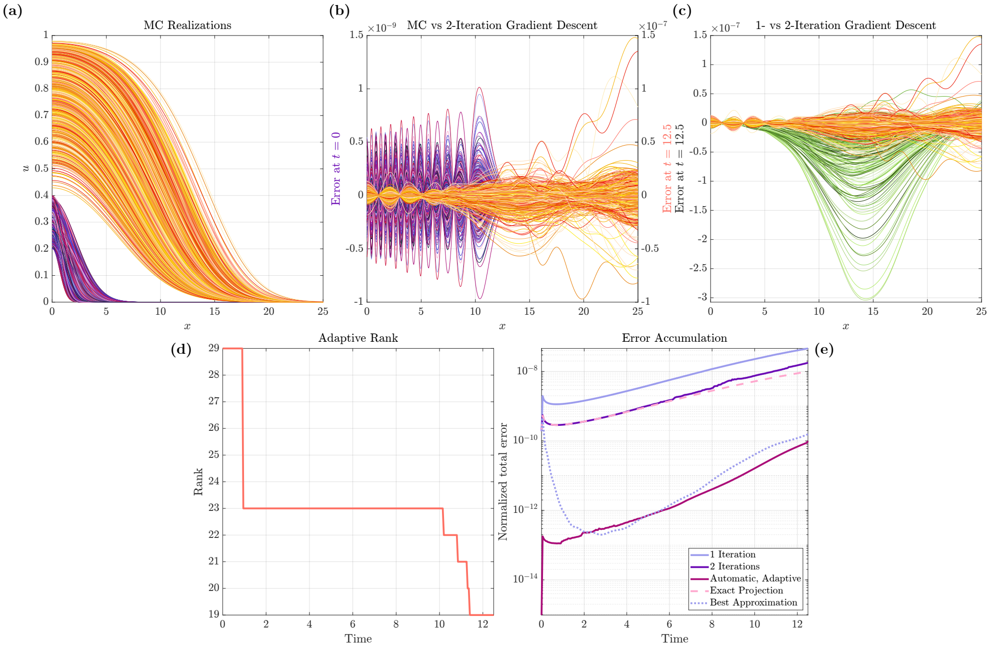

Fig. 8: Stochastic Fisher-KPP PDE (a) Reference solution, the Monte Carlo runs, at two different times. The cool (pink-purple) colors show solution realizations $u(x,t;\omega)$ at the initial conditions $t = 0$, and the warm (yellow-orange) colors show solution realizations at $t = 12.5$. (b) Error of each reconstructed MC realization at the initial and final times; in both instances, errors are centered around zero and much smaller than the solution magnitudes $u(x,t;\omega)$ (error scales at $t = 0$ and $t = 12.5$ differ, and both differ from the scales for $u(x,t;\omega)$). (c) Compares the errors between the 1-iteration (green) and 2-iteration (yellow-orange, as in (b)) gradient-descent retractions at the final time step. (d) Rank as a function of time for the automatic, rank-adaptive integration scheme. (e) Error measured in the Frobenius norm, normalized by the Frobenius norm of the MC solution at final time. From this non-dimensional error, we find that the 2-iteration gradient-descent retraction more closely follows the error incurred by the exact projection. This scheme outperforms the fixed-rank schemes by orders of magnitude.

the first time step; the diffusion term is then treated implicitly and the reaction term explicitly. For the numerical space-time discretization, we use 1,000 grid points in x, 10,001 points in time, and a second-order centered finite difference stencil for the diffusion term. For the stochastic discretization, we use 1,000 MC realizations.

Figure 8 compares the MC run and a rank-15 approximation using our first-order optimal perturbative retraction (2.6a) applied once or twice iteratively each time step. We also compute the solution using our automatic gradient-descent retraction from Algorithm 3.2 with our adaptive retraction from Algorithm 4.1. This scheme is initialized by a rank-29 truncated SVD of the MC initial conditions and has hyperparameters of $r_{\text{inc}} = 5$, $r_{\max} = 30$, $\sigma^* = 10^{-12}$, $\theta^* = 0.05$, $N_{\max} = 8$, and $\Delta^* = 10^{-16}$. In addition, we compare these retractions with the exact projection operator (i.e. the truncated SVD) applied after every time step, which has no projection-retraction error ε_{pr} but does incur dynamical model closure error ε_D (see table 2). We also compute the best possible approximation given by the truncated SVD of the MC solution, which only incurs normal closure error $\varepsilon_{\mathcal{N}}$ (see Table 2). In essence, the exact projection operator is the best approximation we can hope to make with the fixed-rank DLRA because it intrinsically does not have error accumulation. We see that the 2-iteration retraction significantly reduces the error that accumulates over time, but cannot match the best approximation at rank-15. With our robust, adaptive scheme, however, we are able to outperform the best rank-15 approximation despite the inherent error accumulation. The solution rank reduces after the initial conditions, which is reflected in the solution of the adaptive-rank algorithm.

However, the fixed-rank DLRAs already accumulated error from the initial conditions. We also see that the error at $t = 12.5$ seems to be concentrated at large values of x. This may also be remedied in the future by using weighted norms to reduce error in certain locations.

7. Conclusion. We have derived two new sets of retractions, one building off of the other. Optimal perturbative retractions are an improvement over the original perturbative retractions developed in [12]. While we employ a perturbation series in $\overline{U}$ as before, we use the optimal Z matrix, which incorporates higher-order nonlinearities in $\overline{\mathscr{L}}$. The optimal perturbative retractions avoid error associated with overshoot via their optimal choice of coefficients Z given the modes U, which is made possible without pseudoinversion by reorthonormalization. The coefficient update equation (2.2) is equivalent to that used in proper orthogonal decomposition [50, 3] and other reduced-basis methods such as dynamic mode decomposition [53, 56, 60, 36], but here the modes upon which the dynamics are projected are dynamically updated, enabling more accurate model-order reduction without the need for a large offline/training stage. The optimal perturbative retractions are more accurate in approximating the best low-rank approximation than their original counterparts and come at a slightly reduced computational cost. The dominating computational cost is $\mathcal{O}((m+n)r(r+r_L))$, where r_L is the rank of the discretized differential operator $\overline{\mathscr{L}}$; this is the same complexity as the original perturbative retractions [12], the projector-splitting integrator [49], and the randomized singular value decomposition [27].

Gradient-descent retractions are easily generated by repeatedly applying optimal perturbative retractions on the residual between a higher-rank system and our current approximation. If the full-rank system is, in fact, on the low-rank manifold, our gradient-descent retractions converge superlinearly to the best low-rank approximation. Otherwise, we expect linear convergence, but typically the rate of convergence is very fast and increases with the number of iterations applied. Our analysis complements that done for the projector-splitting integrator in the context of fixed-point iteration [34]. The gradient-descent retractions with higher-order perturbative retractions are analogous to higher-order generalizations of Newton's method on the low-rank manifold. Using an adaptive perturbative retraction proved very useful for the gradient-descent retractions since, when far away from the best low-rank approximation, low-order retractions can be more accurate, but as our approximation improves over each iteration, higher-order retractions become more efficient.

We have also shown how to robustify retractions in two ways. First, one may use the Moore-Penrose pseudoinverse for the inversion of the correlation matrix in the high-order optimal perturbative retractions. Second, we can robustify the first-order perturbative retraction by only preserving the span of the modes U. We incorporate this robust retraction into our gradient-descent scheme, obtaining retractions that converge rapidly to the exact projection operator. Pseudoinversion preserves mode continuity, a useful feature for interpretability, but introduces an extra source of error by truncating small singular values. Our latter approach that only preserves the span does not accumulate additional error, but it sacrifices mode continuity. Either scheme may be converted to a rank-adaptive integrator that allows for highly accurate solutions while adapting to the system rank for improved computational efficiency. When we correctly estimate (or underestimate) the rank of the system, the adaptive retraction with gradient-descent converges rapidly to the optimal low-rank approximation. Otherwise, convergence can be slow due to a non-optimal choice of subspace U.

Two novel Dynamically Orthogonal Runge-Kutta families of integrators were derived, the stable, optimal and gradient-descent Dynamically Orthogonal Runge-Kutta schemes, so-DORK and gd-DORK, respectively. The so-DORK schemes build off of the optimal perturbative retractions. By writing the differential operator $\overline{\mathscr{L}}$ as a perturbation series, we update, within the time step, the subspace in which we integrate and in which the system dynamics are projected. Similarly, the gd-DORK schemes update the subspace as we integrate, but they build off of the gradient-descent retractions.

Finally, we verified the efficacy of our retractions and integrators on three examples. We first compared our schemes with the projected Runge-Kutta [32] and projector-splitting [49] integrators on a linear matrix differential equation. Next, we solved an advection-diffusion PDE, again comparing our retractions with those in the literature. We also showed the utility of our rank-adaptive algorithm using the so-DORK integrator. Finally, we used our gradient-descent retractions to build a low-rank implicit-explicit integrator for a nonlinear, stochastic diffusion-reaction PDE. We compared applying one and two iterations of gradient-descent at each time step. The 2-iteration gradient-descent retraction significantly reduced the error accumulation over time. We also compared the fixed-iteration schemes with our automatic algorithm with adaptive rank, which outperforms the fixed-rank methods partially because it is able to stably overapproximate the rank of the solution.

In the future, we believe using weighted norms when we derive our retractions may allow us to strategically reduce errors in regions we care the most about [44, 40, 18]. Another interesting direction is to use an adaptive deterministic or stochastic dimension in which the dimensions of the matrix we approximate changes in time rather than just the rank. To preserve high-order convergence when our low-rank approximation is ill-conditioned, a new approach could involve using an adaptive time step that depends on the condition number of the system. In addition, deriving implicit DORK schemes may prove beneficial for stiff systems. Finally, research into accelerating convergence while increasing the rank of the solution when the rank of $\overline{\mathscr{L}}$ is large is needed.

Acknowledgments. We thank Manan Doshi for his discussion on section 4.1 and the entire MSEAS group for their collaboration.

REFERENCES

[1] P.-A. Absil and I. V. Oseledets, *Low-rank retractions: a survey and new results*, Computational Optimization and Applications, 62 (2015), pp. 5–29.

[2] Ş. T. Beşiktepe, P. F. J. Lermusiaux, and A. R. Robinson, *Coupled physical and biogeochemical data-driven simulations of Massachusetts Bay in late summer: Real-time and post-cruise data assimilation*, Journal of Marine Systems, 40–41 (2003), pp. 171–212, https://doi.org/10.1016/S0924-7963(03)00018-6.

[3] G. Berkooz, P. Holmes, and J. L. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annual review of fluid mechanics, 25 (1993), pp. 539–575.

[4] M. Billaud-Friess, A. Falcó, and A. Nouy, *A geometry based algorithm for dynamical low-rank approximation*, arXiv preprint arXiv:2001.08599, (2020).

[5] M. Billaud-Friess, A. Falcó, and A. Nouy, *A new splitting algorithm for dynamical low-rank approximation motivated by the fibre bundle structure of matrix manifolds*, BIT Numerical Mathematics, (2021), pp. 1–22.

[6] E. Celledoni and B. Owren, *A class of intrinsic schemes for orthogonal integration*, SIAM Journal on Numerical Analysis, 40 (2002), pp. 2069–2084.

[7] G. Ceruti, J. Kusch, and C. Lubich, *A rank-adaptive robust integrator for dynamical low-rank approximation*, BIT Numerical Mathematics, (2022), pp. 1–26.

[8] G. Ceruti and C. Lubich, *An unconventional robust integrator for dynamical low-rank approximation*, BIT Numerical Mathematics, (2021), pp. 1–22.

[9] A. Charous, *High-order retractions for reduced-order modeling and uncertainty quantification*, master's thesis, Massachusetts Institute of Technology, Center for Computational Science and Engineering, Cambridge, Massachusetts, Feb. 2021.

[10] A. Charous, *Dynamical Reduced-Order Models for High-Dimensional Systems*, PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering and Center for Computational Science and Engineering, Cambridge, Massachusetts, June 2023.

[11] A. Charous and P. F. J. Lermusiaux, *Dynamically orthogonal differential equations for stochastic and deterministic reduced-order modeling of ocean acoustic wave propagation*, in OCEANS 2021 IEEE/MTS, IEEE, Sept. 2021, pp. 1–7, https://doi.org/10.23919/OCEANS44145.2021.9705914.

[12] A. Charous and P. F. J. Lermusiaux, *Dynamically Orthogonal Runge–Kutta schemes with perturbative retractions for the dynamical low-rank approximation*, SIAM Journal on Scientific Computing, 45 (2023), pp. A872–A897, https://doi.org/10.1137/21M1431229.

[13] A. Charous and P. F. J. Lermusiaux, *Implicit Dynamically Orthogonal Runge–Kutta schemes and efficient nonlinearity evaluation*, SIAM Journal on Scientific Computing, (2023). To be submitted.

[14] P. A. M. Dirac, *Note on exchange phenomena in the thomas atom*, Mathematical Proceedings of the Cambridge Philosophical Society, 26 (1930), p. 376–385.

[15] M. El-Hachem, S. W. McCue, W. Jin, Y. Du, and M. J. Simpson, *Revisiting the fisher–kolmogorov–petrovsky–piskunov equation to interpret the spreading–extinction dichotomy*, Proceedings of the Royal Society A, 475 (2019), p. 20190378.

[16] F. Feppon and P. F. J. Lermusiaux, *Dynamically orthogonal numerical schemes for efficient stochastic advection and Lagrangian transport*, SIAM Review, 60 (2018), pp. 595–625, https://doi.org/10.1137/16M1109394.

[17] F. Feppon and P. F. J. Lermusiaux, *A geometric approach to dynamical model-order reduction*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), pp. 510–538, https://doi.org/10.1137/16M1095202.

[18] F. Feppon and P. F. J. Lermusiaux, *The extrinsic geometry of dynamical systems tracking nonlinear matrix projections*, SIAM Journal on Matrix Analysis and Applications, 40 (2019), pp. 814–844, https://doi.org/10.1137/18M1192780.

[19] R. A. Fisher, *The wave of advance of advantageous genes*, Annals of eugenics, 7 (1937), pp. 355–369.

[20] B. Gao and P.-A. Absil, *A riemannian rank-adaptive method for low-rank matrix completion*, Computational Optimization and Applications, 81 (2022), pp. 67–90.

[21] S. Gottlieb and C.-W. Shu, *Total variation diminishing runge-kutta schemes*, Mathematics of computation, 67 (1998), pp. 73–85.

[22] P. Grindrod, *The theory and applications of reaction-diffusion equations: patterns and waves*, Clarendon Press, 1996.

[23] A. Gupta, P. J. Haley, D. N. Subramani, and P. F. J. Lermusiaux, *Fish modeling and Bayesian learning for the Lakshadweep Islands*, in OCEANS 2019 MTS/IEEE SEATTLE, Seattle, Oct. 2019, IEEE, pp. 1–10, https://doi.org/10.23919/OCEANS40490.2019.8962892.

[24] A. Gupta and P. F. J. Lermusiaux, *Neural closure models for dynamical systems*, Proceedings of The Royal Society A, 477 (2021), pp. 1–29, https://doi.org/10.1098/rspa.2020.1004.

[25] E. Hairer, *Symmetric projection methods for differential equations on manifolds*, BIT Numerical Mathematics, 40 (2000), pp. 726–734.

[26] E. Hairer, *Solving differential equations on manifolds*, Lec. Notes, Univ. de Geneve, (2011).

[27] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM review, 53 (2011), pp. 217–288.

[28] C. Hauck and S. Schnake, *A predictor-corrector strategy for adaptivity in dynamical low-rank approximations*, arXiv preprint arXiv:2209.00550, (2022).

[29] J. S. Hesthaven, C. Pagliantini, and N. Ripamonti, *Rank-adaptive structure-preserving model order reduction of hamiltonian systems*, ESAIM: Mathematical Modelling and Numerical Analysis, 56 (2022), pp. 617–650.

[30] M. Hochbruck, M. Neher, and S. Schrammer, *Dynamical low-rank integrators for second-order matrix differential equations*, BIT Numerical Mathematics, 63 (2023).

[31] W. Hundsdorfer and J. G. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, vol. 33, Springer Science & Business Media, 2007.

[32] E. Kieri and B. Vandereycken, *Projection methods for dynamical low-rank approximation of high-dimensional problems*, Computational Methods in Applied Mathematics, 19 (2019), pp. 73–92.

[33] O. Koch and C. Lubich, *Dynamical low-rank approximation*, SIAM J. on Matrix Analysis and Applications, 29 (2007), pp. 434–454.

[34] D. Kolesnikov and I. V. Oseledets, *Convergence analysis of projected fixed-point iteration on a low-rank matrix manifold*, Numerical Linear Algebra with Applications, 25 (2018), p. e2140.

[35] A. Kolmogorov, I. Petrovsky, and N. Piscounov, *A study of the diffusion equation with increase in the amount of substance*, Bull. Univ. État Moscou, 1 (1937).

[36] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*, SIAM, 2016.

[37] P. F. J. Lermusiaux, *Error subspace data assimilation methods for ocean field estimation: Theory, validation and applications*, PhD thesis, Harvard University, Cambridge, MA, May 1997.

[38] P. F. J. Lermusiaux, *Data assimilation via Error Subspace Statistical Estimation, part II: Mid-Atlantic Bight shelfbreak front simulations, and ESSE validation*, Monthly Weather Review, 127 (1999), pp. 1408–1432, https://doi.org/10.1175/1520-0493(1999)127⟨1408:DAVESS⟩2.0.CO;2.

[39] P. F. J. Lermusiaux, *Evolving the subspace of the three-dimensional multiscale ocean variability: Massachusetts Bay*, Journal of Marine Systems, 29 (2001), pp. 385–422, https://doi.org/10.1016/S0924-7963(01)00025-2.

[40] P. F. J. Lermusiaux, *Adaptive modeling, adaptive data assimilation and adaptive sampling*, Physica D: Nonlinear Phenomena, 230 (2007), pp. 172–196, https://doi.org/10.1016/j.physd.2007.02.014.

[41] P. F. J. Lermusiaux, *Numerical fluid mechanics*. MIT OpenCourseWare, May 2015, https://ocw.mit.edu/courses/mechanical-engineering/2-29-numerical-fluid-mechanics-spring-2015/lecture-notes-and-references/.

[42] P. F. J. Lermusiaux, P. J. Haley, W. G. Leslie, A. Agarwal, O. Logutov, and L. J. Burton, *Multiscale physical and biological dynamics in the Philippine Archipelago: Predictions and processes*, Oceanography, 24 (2011), pp. 70–89, https://doi.org/10.5670/oceanog.2011.05. Special Issue on the Philippine Straits Dynamics Experiment.

[43] P. F. J. Lermusiaux and A. R. Robinson, *Data assimilation via Error Subspace Statistical Estimation, part I: Theory and schemes*, Monthly Weather Review, 127 (1999), pp. 1385–1407, https://doi.org/10.1175/1520-0493(1999)127⟨1385:DAVESS⟩2.0.CO;2.

[44] P. F. J. Lermusiaux, A. R. Robinson, P. J. Haley, and W. G. Leslie, *Advanced interdisciplinary data assimilation: Filtering and smoothing via error subspace statistical estimation*, in Proceedings of The OCEANS 2002 MTS/IEEE conference, Holland Publications, 2002, pp. 795–802, https://doi.org/10.1109/oceans.2002.1192071.

[45] S. A. Levin, . H. C. Muller-Landau, . R. Nathan, and . J. Chave, *The ecology and evolution of seed dispersal: a theoretical perspective*, Ann. Rev. of Eco., Evol., and Sys., 34 (2003), pp. 575–604.

[46] J. Lin, *Bayesian Learning for High-Dimensional Nonlinear Systems: Methodologies, Numerics and Applications to Fluid Flows*, PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, Sept. 2020.

[47] J. Lin and P. F. J. Lermusiaux, *Minimum-correction second-moment matching: Theory, algorithms and applications*, Numerische Mathematik, 147 (2021), pp. 611–650, https://doi.org/10.1007/s00211-021-01178-8.

[48] C. Lubich, *From quantum to classical molecular dynamics: reduced models and numerical analysis*, European Mathematical Society, 2008.

[49] C. Lubich and I. V. Oseledets, *A projector-splitting integrator for dynamical low-rank approximation*, BIT Numerical Mathematics, 54 (2014), pp. 171–188.

[50] Z. Luo and G. Chen, *Proper orthogonal decomposition methods for partial differential equations*, Academic Press, 2018.

[51] E. Musharbash, F. Nobile, and E. Vidličková, *Symplectic dynamical low rank approximation of wave equations with random parameters*, BIT Numerical Mathematics, 60 (2020), pp. 1153–1201.

[52] C. Pagliantini, *Dynamical reduced basis methods for hamiltonian systems*, Numerische Mathematik, 148 (2021), pp. 409–448.

[53] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Spectral analysis of nonlinear flows*, Journal of fluid mechanics, 641 (2009), pp. 115–127.

[54] T. Ryu, J. P. Heuss, P. J. Haley, Jr., C. Mirabito, E. Coelho, P. Hursky, M. C. Schönau, K. Heaney, and P. F. J. Lermusiaux, *Adaptive stochastic reduced order modeling for autonomous ocean platforms*, in OCEANS 2021 IEEE/MTS, IEEE, Sept. 2021, pp. 1–9, https://doi.org/10.23919/OCEANS44145.2021.9705790.

[55] T. P. Sapsis and P. F. J. Lermusiaux, *Dynamical criteria for the evolution of the stochastic dimensionality in flows with uncertainty*, Physica D: Nonlinear Phenomena, 241 (2012), pp. 60–76, https://doi.org/10.1016/j.physd.2011.10.001.

[56] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, Journal of fluid mechanics, 656 (2010), pp. 5–28.

[57] J. A. Sherratt and J. D. Murray, *Models of epidermal wound healing*, Proceedings of the Royal Society of London. Series B: Biological Sciences, 241 (1990), pp. 29–36.

[58] A. Szlam, A. Tulloch, and M. Tygert, *Accurate low-rank approximations via a few iterations of alternating least squares*, SIAM J. on Matrix Analysis and Applications, 38 (2017), pp. 425–433.

[59] L. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997.

[60] J. H. Tu, *Dynamic mode decomposition: Theory and applications*, PhD thesis, Princeton University, 2013.

[61] M. P. Ueckermann, P. F. J. Lermusiaux, and T. P. Sapsis, *Numerical schemes for dynamically orthogonal equations of stochastic fluid and ocean flows*, Journal of Computational Physics, 233 (2013), pp. 272–294, https://doi.org/10.1016/j.jcp.2012.08.041.

[62] H. Weyl, *Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung)*, Math. Ann., 71 (1912), pp. 441–479.

[63] G. Zhou, W. Huang, K. A. Gallivan, P. Van Dooren, and P.-A. Absil, *A riemannian rank-adaptive method for low-rank optimization*, Neurocomputing, 192 (2016), pp. 72–80.

[64] R. Zimmermann, B. Peherstorfer, and K. Willcox, *Geometric subspace updates with applications to online adaptive nonlinear model reduction*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), pp. 234–261.