SIAM J. SCI. COMPUT.
Vol. 44, No. 3, pp. A1681–A1702
© 2022 Society for Industrial and Applied Mathematics

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

OPTIMIZING OBLIQUE PROJECTIONS FOR NONLINEAR SYSTEMS USING TRAJECTORIES*

SAMUEL E. OTTO†, ALBERTO PADOVAN†, AND CLARENCE W. ROWLEY†

Abstract. Reduced-order modeling techniques, including balanced truncation and $\mathcal{H}_2$-optimal model reduction, exploit the structure of linear dynamical systems to produce models that accurately capture the dynamics. For nonlinear systems operating far away from equilibria, on the other hand, current approaches seek low-dimensional representations of the state that often neglect low-energy features that have high dynamical significance. For instance, low-energy features are known to play an important role in fluid dynamics, where they can be a driving mechanism for shear-layer instabilities. Neglecting these features leads to models with poor predictive accuracy despite being able to accurately encode and decode states. In order to improve predictive accuracy, we propose to optimize the reduced-order model to fit a collection of coarsely sampled trajectories from the original system. In particular, we optimize over the product of two Grassmann manifolds defining Petrov–Galerkin projections of the full-order governing equations. We compare our approach with existing methods including proper orthogonal decomposition, balanced truncation-based Petrov–Galerkin projection, quadratic-bilinear balanced truncation, and the quadratic-bilinear iterative rational Krylov algorithm. Our approach demonstrates significantly improved accuracy both on a nonlinear toy model and on an incompressible (nonlinear) axisymmetric jet flow with 10^5 states.

Key words. model reduction, nonlinear systems, Petrov–Galerkin projection, Riemannian optimization, Grassmann manifold, adjoint sensitivity method

AMS subject classifications. 14M15, 15A03, 34A45, 34C20, 49J27, 49J30, 49M37, 76D55, 90C06, 90C26, 90C45, 93A15, 93C10

DOI. 10.1137/21M1425815

1. Introduction.

Accurate low-dimensional models of physical processes enable a variety of important scientific and engineering tasks to be carried out. However, many real-world systems like complex fluid flows in the atmosphere as well as around and inside aircraft are governed by extremely high-dimensional nonlinear systems—properties that make tasks like real-time forecasting, state estimation, and control computationally prohibitive using the original governing equations. Fortunately, the behavior of these systems is frequently dominated by coherent structures and patterns [14] that may be modeled with equations whose dimension is much smaller [45, 21]. The goal of "reduced-order modeling" is to obtain simplified models that are suitable for forecasting, estimation, and control from the vastly more complicated governing equations provided by physics. For reviews of modern techniques, see [6, 12, 38]. For a striking display of coherent structures in turbulence, see the shadowgraphs in Brown and Roshko [14].

When the system of interest is operating close to an equilibrium point, the governing equations are accurately approximated by their linearization about the equilibrium. In this case, a variety of sophisticated and effective reduced-order modeling techniques can be applied with guarantees on the accuracy of the resulting low-

*Submitted to the journal's Methods and Algorithms for Scientific Computing section June 9, 2021; accepted for publication (in revised form) March 2, 2022; published electronically June 24, 2022.

https://doi.org/10.1137/21M1425815

Funding: The work of the first author was supported by the NSF through grant DGE-2039656. This work was supported by the Army Research Office under grant W911NF-17-1-0512 and by the Air Force Office of Scientific Research under grant FA9550-19-1-0005.

†Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08540 USA (sotto@princeton.edu, apadovan@princeton.edu, cwrowley@princeton.edu).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

dimensional model [4, 12]. Put simply, linearity provides an elegant and complete characterization of the system's trajectories in response to inputs, disturbances, and initial conditions that can be exploited to build simplified models whose trajectories closely approximate the ones from the original system. For instance, the balanced truncation method introduced by Moore [33] yields a low-dimensional projection of the original system that simultaneously retains the most observable and controllable states of the system and provides bounds on various measures of reduced-order model error [4]. A computationally efficient approximation called balanced proper orthogonal decomposition (BPOD) [37] is suitable for high-dimensional fluid flow applications. Another approach is to find a reduced-order model that is as close as possible to a stable full-order model (FOM) with respect to the $\mathcal{H}_2$ norm. Algorithms like the iterative rational Krylov algorithm (IRKA) [20] are based on satisfying necessary conditions for $\mathcal{H}_2$-optimality.

Various generalizations of linear model reduction techniques have also been developed for bilinear [6, 8, 19], quadratic-bilinear [9, 11, 10], and lifted nonlinear systems [27] based on truncated Volterra series expansion of the output. These methods extend the region of validity for reduced-order models about stable equilibria, yet still suffer as high-order nonlinearities become dominant far away from an equilibrium. These techniques also require solutions of large-scale Sylvester or Lyapunov equations, making them difficult to apply to fluid flows whose state dimensions can easily exceed 10^5.

One commonality among the above model reduction approaches is that they utilize oblique projections to retain coordinates or "features" with high variance or "energy" as well as any coordinates with low variance that significantly influence the dynamics at future times [12, 37]. These small but dynamically significant features are known to play an important role in driving the growth of instabilities in "shear flows" such as mixing layers and jets. Linearizations of these shear flows often result in nonnormal systems, which can exhibit large transient growth in response to low-energy perturbations [46, 43]. Some successful approaches [5, 3, 23, 24] have involved oblique projections of the nonlinear dynamics onto subspaces identified from the dynamics linearized about an equilibrium. However, this approach is often not satisfactory since the linearized dynamics become inaccurate as the state moves away from the equilibrium and nonlinear effects become significant. In this paper we illustrate how such nonlinear effects can cause reduced-order models obtained using the various approaches described above to perform poorly, for instance, on a simple three-dimensional system as well as on a high-dimensional axisymmetric jet flow.

When dealing with nonlinear systems operating far away from equilibria, nonlinear model reduction approaches tend to follow a two-step process: first identify a set, typically a smooth manifold or a subspace, near which the state of the system is known to lie, then model the dynamics in this set either by a projection of the governing equations or by a black-box data-driven approach. The most common approach to identify a candidate subspace is proper orthogonal decomposition (POD), whose application to the study of complex fluid flows was pioneered by Lumley [30]. The dynamics may also be projected onto nonlinear manifolds using "nonlinear Galerkin" methods [31, 35]. Recently, more sophisticated manifold-learning techniques like deep convolutional autoencoders have also been used [29]. The main obstacle encountered by the manifold-learning and POD-based approaches is that they neglect coordinates with low variance even if they are important for correctly forecasting the system's dynamics. For instance, in our jet flow example, we find that a model with 50 POD modes capturing 99.6% of the state's variance still yields poor predictions that diverge from the FOM.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

In order to identify and retain the dynamically important coordinates while remaining tractable for very large-scale systems like fluid flows, we shall optimize an oblique projection operator to minimize the prediction error of the corresponding reduced-order model on a collection of sampled trajectories. In this framework, oblique projection operators of a fixed dimension are identified with pairs of subspaces in Grassmann manifolds that meet a transversality condition. We show that the pairs of subspaces defining oblique projection operators are open, dense, and connected in the product of Grassmann manifolds, and we prove that solutions of our optimization problem exist when it is appropriately regularized. Optimization is carried out using the Riemannian conjugate gradient algorithm introduced by Sato [39] with formulas for the exponential map and parallel translation along geodesics given by Edelman, Arias, and Smith [18]. We provide mild conditions under which the algorithm is guaranteed to converge to a locally optimal oblique projection operator.

Related techniques based on optimizing projection subspaces have been used to produce $\mathcal{H}_2$-optimal reduced-order models for linear and bilinear systems. Most approaches focus on optimizing orthogonal projection operators over a single Grassmann manifold [50, 42, 25] or an orthogonal Stiefel manifold [51, 42, 47, 52, 49]. On the other hand, an alternating minimization technique over the two Grassmann manifolds defining an oblique projection is proposed T. Zeng and Lu [53] for $\mathcal{H}_2$-optimal reduction of linear systems. For systems with quadratic nonlinearities, Jiang and Xu [25] present an approach to optimize orthogonal projection operators based on the same truncated generalization of the $\mathcal{H}_2$ norm used by Benner, Goyal, and Gugercin [11]. Our approach differs from the ones mentioned above in that it may be used to find optimal reduced-order models based on oblique projections for general nonlinear high-dimensional systems based on sampled trajectories.

2. Projection-based reduced-order models. Consider a physical process, modeled by an input-output dynamical system

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, x &= f(x,u), \qquad x(t_0) = x_0, \\ y &= g(x), \end{aligned} \tag{2.1}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^d$, and output y in $\mathbb{R}^m$, each space being equipped with the Euclidean inner product. We shall often refer to (2.1) as the FOM. Our goal is to use one or more discrete-time histories of observations $y_l = y(t_l)$ at sample times $t_0 < \cdots < t_{L-1}$ in order to learn the key dynamical features of (2.1) and produce a reduced-order model that captures these effects. Throughout the paper we assume the following.

Assumption 2.1. The functions $(x,t) \mapsto f(x,u(t))$ and $x \mapsto g(x)$ in (2.1) have continuous partial derivatives with respect to x up to second order.

We shall use our observation data to learn an r-dimensional subspace V of $\mathbb{R}^n$ in which to represent the state of the system (2.1). Since $f(x,u)$ might not lie in V when $x \in V$, we shall also find another r-dimensional subspace W of $\mathbb{R}^n$ with $V \oplus W^\perp = \mathbb{R}^n$ in order to construct an oblique projection operator $P_{V,W} : \mathbb{R}^n \to \mathbb{R}^n$ satisfying

$$P_{V,W}\, x \in V \quad \text{and} \quad x - P_{V,W}\, x \in W^\perp \qquad \forall x \in \mathbb{R}^n. \tag{2.2}$$

Every rank-r oblique projection operator can be identified with a pair of subspaces (V,W) satisfying $V \oplus W^\perp = \mathbb{R}^n$ (see section 5.9 of Meyer [32]), and we denote the set of such subspaces by $\mathcal{P}$. Moreover, if $\Phi, \Psi \in \mathbb{R}^{n\times r}$ are matrices with $V = \operatorname{Range}\Phi$ and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$W = \operatorname{Range}\Psi$, then it is easily shown that $(V, W) \in \mathcal{P}$ if and only if $\det(\Psi^T\Phi) \neq 0$, and the corresponding projection operator is given explicitly by

$$P_{V,W} = \Phi(\Psi^T\Phi)^{-1}\Psi^T. \tag{2.3}$$

Applying the projection defined by $(V, W) \in \mathcal{P}$ to the FOM (2.1), we obtain a Petrov–Galerkin reduced-order model whose state $\hat{x} \in V$ evolves according to

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\hat{x} = P_{V,W}f(\hat{x}, u), \qquad \hat{x}(0) = P_{V,W}x_0 \tag{2.4}$$

with observations given by $\hat{y} = g(\hat{x})$. The two subspaces V, W uniquely define the projection $P_{V,W}$ and the reduced-order model (2.4). With the initial condition x_0 and input signal u fixed, the output of the reduced-order model at each sample time $\hat{y}_l(V, W) = \hat{y}(t_l; (V, W))$ is a function of the chosen subspaces V, W.

Let $L_y : \mathbb{R}^m \to [0, +\infty)$ be a smooth penalty function for the difference between each observation y_l and the model's prediction $\hat{y}_l(V, W)$. Let us also introduce a smooth nonnegative-valued function $\rho(V, W)$, to be defined precisely in section 3, that will serve as regularization by preventing minimizing sequences of subspaces (V, W) from approaching points outside the set $\mathcal{P}$ in which valid Petrov–Galerkin projections can be defined. Using this regularization with a weight $\gamma > 0$ allows us to seek a minimum of the cost defined by

$$J(V, W) = \frac{1}{L}\sum_{l=0}^{L-1} L_y\left(\hat{y}_l(V, W) - y_l\right) + \gamma\rho(V, W) \tag{2.5}$$

over all pairs of r-dimensional subspaces (V, W), subject to the reduced-order dynamics (2.4). Here we shall consider the case when there is a single trajectory generated from a known initial condition since it will be easy to handle multiple trajectories from multiple known initial conditions once we understand the single trajectory case. The cost function (2.5) defines an optimization problem, and in the following section we define a suitable regularization function ρ and develop a technique for iteratively solving this problem. We refer to this approach for constructing reduced-order models as trajectory-based optimization for oblique projections (TrOOP).

Remark 2.2 (integrated objectives and $\mathcal{H}_2$-optimal model reduction). We may also optimize a cost function where the sum in (2.5) is replaced by an integral approximated using numerical quadrature; the details are given in SM4.1. When the FOM (2.1) is a stable linear-time-invariant system and the trajectories $y(t)$ are generated by unit impulse responses from each input channel, then the $\mathcal{H}_2$ norm [4] of the difference between the reduced-order model and the FOM can be written as a sum of integrated objectives $\int_0^\infty \|\hat{y}(t; (V, W)) - y(t)\|_2^2\,\mathrm{d}\,t$. After approximating these integrals by integrals over finite time-horizons, we may employ the technique described in SM4.1 for $\mathcal{H}_2$-optimal model reduction.

3. Optimization domain, representatives, and regularization. The set containing all r-dimensional subspaces of $\mathbb{R}^n$ can be endowed with the structure of a compact Riemannian manifold called the Grassmann manifold, which has dimension $nr - r^2$ and is denoted $\mathcal{G}_{n,r}$. Therefore, our optimization problem entails minimizing the cost given by (2.5) over the subset $\mathcal{P}$ of the product manifold $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ on which oblique projection operators are defined. The goal of this section will be to characterize the topology of the set $\mathcal{P}$ and to introduce an appropriate regularization

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

function ρ so that we may instead consider the unconstrained minimization of (2.5) over $\mathcal{M}$. We also describe how to work with matrix representatives of the relevant subspaces that can be stored in a computer.

3.1. Grassmann manifold and representatives of subspaces. First we describe some basic properties of the Grassmann manifold that can be found in [1, 2, 7]. If $\mathbb{R}_*^{n,r}$ denotes the smooth manifold of $n \times r$ matrices with linearly independent columns, then $\mathcal{G}_{n,r}$ can be identified with the quotient manifold of $\mathbb{R}_*^{n,r}$ under the action of the general linear group GL_r defining changes of basis $GL_r \times \mathbb{R}_*^{n,r} \to \mathbb{R}_*^{n,r}$: $(M, X) \mapsto XM$. Since this group action is free and proper, the quotient manifold theorem (Theorem 21.10 in [28]) ensures that $\mathbb{R}_*^{n,r}/GL_r$ is a smooth manifold and the quotient map sending $X \in \mathbb{R}_*^{n,r}$ to its equivalence class in $\mathbb{R}_*^{n,r}/GL_r$,

$$[X] = \{Y \in \mathbb{R}_*^{n,r} \ : \ Y = XM \text{ for some } M \in GL_r \}, \tag{3.1}$$

is a smooth submersion. Each subspace $\operatorname{Range} X \in \mathcal{G}_{n,r}$ is identified with the equivalence class $[X] \in \mathbb{R}_*^{n,r}/GL_r$.

In order to optimize the pairs of abstract subspaces $(V, W) \in \mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ defining oblique projections, we work with the matrix representative of these subspaces in the so-called structure space $\bar{\mathcal{M}} = \mathbb{R}_*^{n,r} \times \mathbb{R}_*^{n,r}$. A pair of matrices $(\Phi, \Psi) \in \bar{\mathcal{M}}$ are representatives of $(V, W) \in \mathcal{M}$ if $V = \operatorname{Range} \Phi$ and $W = \operatorname{Range} \Psi$. The "canonical projection" map $\pi : \bar{\mathcal{M}} \to \mathcal{M}$ is defined by

$$\pi : (\Phi, \Psi) \mapsto (\operatorname{Range} \Phi, \operatorname{Range} \Psi), \tag{3.2}$$

and it is clear that the set of all representatives of $(V, W) \in \mathcal{M}$ is given by the preimage set $\pi^{-1}(V, W)$. The canonical projection map is a surjective submersion since its component maps $\Phi \mapsto \operatorname{Range} \Phi$ and $\Psi \mapsto \operatorname{Range} \Psi$ are surjective submersions. Consequently, Theorem 4.29 in Lee [28] provides the useful result that a function $F : \mathcal{M} \to \mathcal{N}$, with $\mathcal{N}$ another smooth manifold, is smooth if and only if $F \circ \pi$ is smooth.

Suppose that $(V, W) \in \mathcal{P}$ is a pair of subspaces that define an oblique projection and $(\Phi, \Psi) \in \pi^{-1}(V, W)$ is a choice of representatives. We observe that the oblique projection operator given explicitly by (2.3) is independent of the choice of representatives—as it should be, given that $P_{V,W}$ is uniquely defined in terms of abstract subspaces alone. Using the representatives and an r-dimensional state z defined by $\hat{x} = \Phi z \in V$, we obtain a representative of the reduced-order model (2.4) given by

$$\boxed{\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, z &= (\Psi^T\Phi)^{-1}\Psi^T f(\Phi z, u) =: \tilde{f}\,(z, u;\ (\Phi, \Psi)), \quad z(t_0) = (\Psi^T\Phi)^{-1}\Psi^T x_0, \\ \hat{y} &= g(\Phi z) =: \tilde{g}\,(z;\ (\Phi, \Psi)) \end{aligned}} \tag{3.3}$$

that can be simulated on a computer. While the state $z(t)$ of (3.3) depends on the choice of $(\Phi, \Psi) \in \pi^{-1}(V, W)$, the output $\hat{y}(t)$ depends only on the subspaces (V, W) and not on our choice of matrix representatives.

Consequently, any function of (Φ, Ψ) that depends only on the output $\hat{y}(t)$ of (3.3) can be viewed as a function on $\mathcal{M}$ composed with the canonical projection π. Hence, we can evaluate our cost function (2.5) for a subspace pair (V, W) by computing

$$\bar{J}(\Phi, \Psi) = J(\pi(\Phi, \Psi)) \tag{3.4}$$

for any choice of represenatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$, that is, by evaluating the sum in (2.5) using the output $\hat{y}(t)$ generated by (3.3). Moreover, Theorem 4.29 in Lee [28] tells us that J is smooth if and only if $\bar{J}$ is smooth.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

3.2. Topology of the optimization problem domain. The main result of this section is the following.

THEOREM 3.1 (topology of subspaces that define oblique projections). *Let $\mathcal{P}$ denote the pairs of subspaces $(V, W) \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ such that $V \oplus W^\perp = \mathbb{R}^n$. Then $\mathcal{P}$ is open, dense, and connected in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. Moreover, $\mathcal{P}$ is diffeomorphic to the set of rank-r projection operators*

$$\mathbb{P} = \left\{ P \in \mathbb{R}^{n \times n} \ : \ P^2 = P \quad \textit{and} \quad \operatorname{rank}(P) = r \right\}. \tag{3.5}$$

Proof. See SM1. ∎

The openness of $\mathcal{P}$ in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ means that it is a submanifold of $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ with the same dimension $\dim \mathcal{P} = 2nr - 2r^2$. The connectedness result is especially important since it means than an optimization routine can access any point in the set $\mathcal{P}$ by a smooth path from any initial guess without ever encountering the "bad set" $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r} \setminus \mathcal{P}$. In other words the bad set doesn't cut off access to any region of $\mathcal{P}$ by an optimizer that progresses along a smooth path, e.g., a gradient flow.

The reduced-order model (2.4) may not have a solution over the desired time interval $[t_0, t_{L-1}]$ for every projection operator defined by $(V, W) \in \mathcal{P}$. The following result characterizes the appropriate domain $\mathcal{D} \subset \mathcal{P}$ over which the reduced-order model has a unique solution as well as the key properties of solutions when they exist.

PROPOSITION 3.2 (properties of reduced-order model solutions). *When the reduced-order model* (2.4) *has a solution over the time interval $[t_0, t_{L-1}]$, it is unique. Let $\mathcal{D} \subset \mathcal{P}$ denote the set of subspace pairs (V, W) for which the resulting reduced-order model* (2.4) *has a unique solution over the time interval $[t_0, t_{L-1}]$, and let $\hat{x}(t; (V, W))$ denote the state of* (2.4) *with $(t, (V, W)) \in [t_0, t_{L-1}] \times \mathcal{D}$. Then*

1. *$\mathcal{D}$ is open in $\mathcal{P}$, and hence $\mathcal{D}$ is also open in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$;*
2. *when $\frac{\partial}{\partial x} f(x, u(t))$ is bounded, then $\mathcal{D} = \mathcal{P}$;*
3. *if $(x, t) \mapsto f(x, u(t))$ has continuous partial derivatives with respect to x up to order $k \geq 1$, then $(t, (V, W)) \mapsto \hat{x}(t; (V, W))$ is continuously differentiable with respect to (V, W) up to order k on $[t_0, t_{L-1}] \times \mathcal{D}$;*
4. *if $\{(V_k, W_k)\}_{k=1}^{\infty} \subset \mathcal{D}$ is a sequence approaching $(V_k, W_k) \to (V_0, W_0) \in \mathcal{P} \setminus \mathcal{D}$ and $\hat{x}(t; (V_k, W_k))$ are the corresponding solutions of* (2.4)*, then*

$$\max_{t \in [t_0, t_{L-1}]} \|\hat{x}(t; (V_k, W_k))\| \to \infty \quad \textit{as} \quad k \to \infty. \tag{3.6}$$

Proof. The claims follow from standard results in the theory of ODEs that can be found in Kelly and Peterson [26]. We give the detailed proof in SM2. ∎

In particular, Proposition 3.2 shows that the solutions produced by the reduced-order model are twice continuously differentiable over $\mathcal{D}$ and blow up as points outside of $\mathcal{D}$ are approached. In the special case when the governing equations (2.1) have a bounded Jacobian, we may dispense with $\mathcal{D}$ entirely since the projection-based reduced-order model always has a unique solution.

3.3. Regularization and existence of a minimizer. Without regularization, we cannot guarantee a priori that a sequence of subspace pairs with decreasing cost doesn't approach a point outside of the set $\mathcal{P}$ where projection operators are defined. That is, a minimizer for the cost function (2.5) may not even exist in $\mathcal{P}$, in which case our optimization problem would have no solution. In order to address this issue, we

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

introduce a regularization function $\rho(V,W)$ into the cost (2.5) that "blows up" to $+\infty$ as the subspaces (V,W) approach any point outside of $\mathcal{P}$ and nowhere else. In order to do this, we use the fact that $(V,W) \in \mathcal{P}$ if and only if all matrix representatives $(\Phi,\Psi) \in \pi^{-1}(V,W)$ have $\det(\Psi^T\Phi) \neq 0$. While this condition characterizes the set $\mathcal{P}$, we cannot use $\det(\Psi^T\Phi)$ directly since its nonzero value depends on the choice of representatives. But this problem is easily solved by an appropriate normalization, leading us to define the regularization of (2.5) in terms of representatives according to

$$\boxed{\rho \circ \pi(\Phi,\Psi) = -\log\left(\frac{\det(\Psi^T\Phi)^2}{\det(\Phi^T\Phi)\det(\Psi^T\Psi)}\right).} \tag{3.7}$$

We observe that the function $\rho : \mathcal{P} \to \mathbb{R}$ in (3.7) is well defined because $\rho\circ\pi(\Phi,\Psi)$ does not depend on the representatives (Φ,Ψ) thanks to the product rule for determinants.

The following theorem shows that the regularization defined by (3.7) has the desirable properties that it vanishes when $V = W$ and "blows up" as (V,W) escapes the set $\mathcal{P}$. When $V = W$, the resulting projection operator $P_{V,V}$ is the orthogonal projection onto V.

THEOREM 3.3 (regularization). *The minimum value of ρ defined by* (3.7) *over $\mathcal{P}$ is zero, and this minimum value $\rho(V,W) = 0$ is attained if and only if $V = W$. On the other hand, if $(V_0,W_0) \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r} \setminus \mathcal{P}$ and $\{(V_n,W_n)\}_{n=1}^{\infty}$ is a sequence of subspaces in $\mathcal{P}$ such that $(V_n,W_n) \to (V_0,W_0)$ as $n \to \infty$, then $\lim_{n\to\infty} \rho(V_n,W_n) = \infty$.*

Proof. See SM3. ∎

We must also rule out the possibility that a sequence of subspace pairs with decreasing cost approaches a point where the reduced-order model does not have a unique solution. By Proposition 3.2, we do not have this problem when the FOM has a bounded Jacobian since the reduced-order model always has a unique solution, i.e., $\mathcal{D} = \mathcal{P}$. On the other hand, when $\mathcal{D} \neq \mathcal{P}$ we may accomplish this by choosing a cost function that blows up if the states of the reduced-order model blow up. In particular, we assume the following.

Assumption 3.4. Let $\mathcal{D}$ be as in Proposition 3.2 and $\mathcal{P}$ be the subset of $(V,W) \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ for which $V \oplus W^{\perp} = \mathbb{R}^n$. If $\mathcal{D} \neq \mathcal{P}$ and $\{(V_k,W_k)\}_{k=1}^{\infty} \subset \mathcal{D}$ is any sequence producing solutions $\hat{x}(t;(V_k,W_k))$ of the reduced-order model (2.4) such that

$$\max_{t\in[t_0,t_{L-1}]} \|\hat{x}(t;(V_k,W_k))\| \to \infty \quad \text{as} \quad k \to \infty, \tag{3.8}$$

then we assume that $J(V_k,W_k) \to \infty$.

In practice, this is a reasonable assumption if $g(x) \to \infty$ as $\|x\| \to \infty$ and $L_y(y) \to \infty$ as $\|y\| \to \infty$. Alternatively, one could add a new regularization term to the cost function (2.5) that penalizes reduced-order model states with large magnitudes. In Corollary SM3.1 we show that a minimizer of the cost function (2.5) exists in the valid set $\mathcal{D} \subset \mathcal{P}$ when Assumption 3.4 holds, and we use the regularization described by (3.7) with any positive weight $\gamma > 0$.

4. Optimization algorithm. In this section we describe how to optimize the projection subspaces by minimizing the cost function (2.5) over the product of Grassmann manifolds $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ using the Riemannian conjugate gradient algorithm described by Sato in [39]. We use the exponential map and parallel translation along geodesics given by Edelman, Arias, and Smith [18], and we provide an adjoint sensitivity method for computing the gradient of the cost function. Other geometric optimization algorithms such as stochastic gradient descent [13, 41] and quasi-Newton

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

methods [36, 22] are also well suited for high-dimensional problems and rely on the same key ingredients we provide here.

4.1. Computing the gradient. In order to compute the gradient we endow $\mathcal{G}_{n,r}$ with a Riemannian metric, and we use the product metric on $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. This allows us to perform key operations such as constructing geodesics and parallel translates of tangent vectors on $\mathcal{M}$ by treating its two components separately [28]. We follow Absil, Mahony, and Sepulchre [2], whereby the metric on $\mathcal{G}_{n,r}$ is induced by a compatible metric on $\mathbb{R}_*^{n\times r}$ acting on lifted representatives of tangent vectors in a prescribed "horizontal space." The Riemannian metric we adopt for the structure space $\bar{\mathcal{M}} = \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$ is given by the product metric

$$\langle (X_1, Y_1),\ (X_2, Y_2)\rangle_{(\Phi,\Psi)} = \underbrace{\operatorname{Tr}\left[(\Phi^T\Phi)^{-1}X_1^T X_2\right]}_{\langle X_1,\ X_2\rangle_\Phi} + \underbrace{\operatorname{Tr}\left[(\Psi^T\Psi)^{-1}Y_1^T Y_2\right]}_{\langle Y_1,\ Y_2\rangle_\Psi}. \tag{4.1}$$

The gradient, expressed in terms of lifted representatives, is then found by computing the gradient with respect to the matrix representatives in the structure space.

For any tangent vector $\xi \in T_p\mathcal{M}$ and representative $\bar{p} \in \bar{\mathcal{M}}$ such that $p = \pi(\bar{p})$, there is an infinite number of possible $\bar{\xi} \in T_{\bar{p}}\bar{\mathcal{M}}$ that could serve as representatives of ξ in the sense that $\xi = \mathrm{D}\,\pi(\bar{p})\bar{\xi}$. A unique representative of ξ is identified by observing that the preimage $\pi^{-1}(p)$ of any $p \in \mathcal{M}$ is a smooth submanifold of $\bar{\mathcal{M}}$ yielding a decomposition of the tangent space $T_{\bar{p}}\bar{\mathcal{M}}$ into a direct sum of the "vertical space" defined by $\mathcal{V}_{\bar{p}} = T_{\bar{p}}\pi^{-1}(p)$ and the "horizontal space" defined as its orthogonal complement $\mathcal{H}_{\bar{p}} = \mathcal{V}_{\bar{p}}^{\perp}$. The horizontal and vertical spaces for a product manifold are the products of the horizontal and vertical spaces for each component in the Cartesian product, and we have

$$\mathcal{V}_\Phi = \left\{\Phi A \ :\ A \in \mathbb{R}^{r\times r}\right\}, \qquad \mathcal{H}_\Phi = \left\{X \in \mathbb{R}^{n,r} \ :\ \Phi^T X = 0\right\}, \qquad \Phi \in \mathbb{R}_*^{n,r}. \tag{4.2}$$

Using the horizontal distribution on the structure space, we have the following.

Definition 4.1 (horizontal lift [2]). *Given $\xi \in T_p\mathcal{M}$ and a representative $\bar{p} \in \pi^{-1}(p)$, the "horizontal lift" of ξ is the unique element $\bar{\xi}_{\bar{p}} \in \mathcal{H}_{\bar{p}}$ such that $\xi = \mathrm{D}\,\pi(\bar{p})\bar{\xi}_{\bar{p}}$.*

The horizontal lifts of a tangent vector $\xi \in T_V\mathcal{G}_{n,r}$ to either of the component Grassmann manifolds at different representatives transform according to

$$\bar{\xi}_{\Phi S} = \bar{\xi}_\Phi S \in \mathbb{R}^{n\times r} \qquad \forall S \in GL_r \tag{4.3}$$

for every $\Phi \in \mathbb{R}_*^{n,r}$ with $\operatorname{Range}\Phi = V$, as shown by Example 3.6.4 in [2]. The structure space metric (4.1) induces a Riemannian product metric on $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ defined in terms of horizontal lifts

$$\langle (\xi_1, \zeta_1),\ (\xi_2, \zeta_2)\rangle_{(V,W)} = \underbrace{\operatorname{Tr}\left[(\Phi^T\Phi)^{-1}(\bar{\xi}_{1\Phi})^T\bar{\xi}_{2\Phi}\right]}_{\langle \xi_1,\ \xi_1\rangle_V} + \underbrace{\operatorname{Tr}\left[(\Psi^T\Psi)^{-1}(\bar{\zeta}_{1\Psi})^T\bar{\zeta}_{2\Psi}\right]}_{\langle \zeta_1,\ \zeta_2\rangle_W}. \tag{4.4}$$

This metric is independent of the choice of representatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$ thanks to the transformation property (4.3).

An important consequence of the orthogonality of the horizontal and vertical subspaces is that the horizontal lift of the gradient of the cost function $J : \mathcal{M} \to \mathbb{R}$ is given by the gradient of $\bar{J} = J \circ \pi : \bar{\mathcal{M}} \to \mathbb{R}$ [2]; that is,

$$\overline{\operatorname{grad} J(V, W)}_{(\Phi,\Psi)} = \operatorname{grad}\bar{F}(\Phi, \Psi) \qquad \forall (\Phi, \Psi) \in \pi^{-1}(V, W). \tag{4.5}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

This means that the gradient computed with respect to the matrix representatives in the structure space is the appropriate lifted representative in the horizontal space at (Φ, Ψ) of the gradient tangent to $\mathcal{M}$ at (V, W). The gradient of the lifted cost function $\bar{J}$ can be computed using the adjoint sensitivity method described below in Theorem 4.2. The analogous adjoint sensitivity method for a cost function in which the error is integrated over time, rather than being summed over $\{t_l\}_{l=0}^{L-1}$ as in (2.5), is provided by Theorem SM4.1.

THEOREM 4.2 (gradient with respect to model parameters). *Suppose we have observation data $\{y_1, \dots, y_L\}$ generated by the FOM (2.1) at sample times $t_0 < \cdots < t_{L-1}$ with initial condition x_0 and input signal u. Consider the reduced-order model representative (3.3) with parameters $\theta = (\Phi, \Psi)$ in the structure space $\bar{\mathcal{M}}$, which is a Riemannian manifold. With $\pi(\theta) \in \mathcal{D}$, let $\hat{y}_i(\theta) = \hat{y}(t_i; \theta)$ be the observations at the corresponding times t_i generated by (3.3), and let $z(t; \theta)$ denote the state trajectory of (3.3). Then the cost function*

$$\bar{J}(\theta) := \sum_{i=0}^{L-1} L_y(\hat{y}_i(\theta) - y_i), \tag{4.6}$$

measuring the error between the observations generated by the models, is differentiable at every $\theta \in \pi^{-1}(\mathcal{D})$. Let

$$F(t) = \frac{\partial \tilde{f}}{\partial z}(z(t;\theta), u(t);\ \theta) \qquad : \mathbb{R}^r \to \mathbb{R}^r, \tag{4.7a}$$

$$S(t) = \frac{\partial \tilde{f}}{\partial \theta}(z(t;\theta), u(t);\ \theta) \qquad : T_\theta \bar{\mathcal{M}} \to \mathbb{R}^r, \tag{4.7b}$$

$$H(t) = \frac{\partial \tilde{g}}{\partial z}(z(t;\theta);\ \theta) \qquad : \mathbb{R}^r \to \mathbb{R}^m, \tag{4.7c}$$

$$T(t) = \frac{\partial \tilde{g}}{\partial \theta}(z(t;\theta);\ \theta) \qquad : T_\theta \bar{\mathcal{M}} \to \mathbb{R}^m, \tag{4.7d}$$

denote the linearized dynamics and observation functions around $z(t; \theta)$, let $g_i = \operatorname{grad} L_y(\hat{y}_i(\theta) - y_i)$, and define an adjoint variable $\lambda(t)$ that satisfies

$$-\frac{\mathrm{d}}{\mathrm{d}\,t}\lambda(t) = F(t)^* \lambda(t), \qquad t \in (t_i, t_{i+1}], \qquad 0 \le i < L-1, \tag{4.8a}$$

$$\lambda(t_i) = \lim_{t \to t_i^+} \lambda(t) + H(t_i)^* g_i, \tag{4.8b}$$

$$\lambda(t_{L-1}) = H(t_{L-1})^* g_{L-1}. \tag{4.8c}$$

Here $(\cdot)^*$ denotes the adjoint of a linear operator with respect to the inner products on the appropriate spaces. Then the gradient of the cost function (4.6) is given by

$$\boxed{\operatorname{grad} \bar{J}(\theta) = \left(\frac{\partial z}{\partial \theta}(t_0; \theta)\right)^* \lambda(t_0) + \int_{t_0}^{t_{L-1}} S(t)^* \lambda(t)\ \mathrm{d}\,t + \sum_{i=0}^{L-1} T(t_i)^* g_i.} \tag{4.9}$$

Proof. See SM4. ∎

The explicit form of each term required to compute the horizontal lift of the gradient of the cost function (2.5) using Theorem 4.2 is provided by the following Proposition 4.3. In order to simplify these expressions, we work with orthonormal

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

representatives, i.e., $(\Phi, \Psi) \in \pi^{-1}(V, W)$ such that $\Phi^T\Phi = \Psi^T\Psi = I_r$ together with the additional condition $\det(\Psi^T\Phi) > 0$. Such representatives can always be obtained via QR-factorization and adjusting the sign of a column of Φ or Ψ. The horizontal lift of the gradient computed at any other representatives $(\Phi S, \Psi T)$ with $S, T \in GL_r$ can be obtained from the horizontal lift of the gradient computed at (Φ, Ψ) via (4.3).

PROPOSITION 4.3 (required terms for gradient). *We assume that the representatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$ have been chosen such that $\Phi^T\Phi = \Psi^T\Psi = I_r$ and $\det(\Psi^T\Phi) > 0$, and we let $A = (\Psi^T\Phi)^{-1}$. Then the terms required to compute the gradient of the cost function using the model* (3.3) *with respect to the representatives in the structure space via Theorem* 4.2 *are given by*

$$F(t)^* = \left(\frac{\partial \tilde{f}}{\partial z}\big(z(t), u(t)\big)\right)^T, \tag{4.10}$$

$$S(t)^* v = \Big(\left(\frac{\partial f}{\partial x}\big(\Phi z(t), u(t)\big)\right)^T \Psi A^T v z(t)^T - \Psi A^T v \tilde{f}(z(t), u(t))^T, \left(f\big(\Phi z(t), u(t)\big) - \Phi \tilde{f}\big(z(t), u(t)\big)\right) v^T A \Big) \qquad \forall v \in \mathbb{R}^r, \tag{4.11}$$

$$H(t)^* = \left(\frac{\partial \tilde{g}}{\partial z}\big(z(t)\big)\right)^T, \tag{4.12}$$

$$T(t)^* w = \left(\left(\frac{\partial g}{\partial x}\big(\Phi z(t)\big)\right)^T w z(t)^T,\ 0 \right) \qquad \forall w \in \mathbb{R}^m, \tag{4.13}$$

$$\left(\frac{\partial z}{\partial(\Phi, \Psi)}\big(t_0; (\Phi, \Psi)\big)\right)^T v = \Big(- \Psi A^T v z(t_0)^T,\ \big(x_0 - \Phi z(t_0)\big) v^T A \Big) \qquad \forall v \in \mathbb{R}^r. \tag{4.14}$$

The gradient of the regularization function (3.7) *is given by*

$$\operatorname{grad}(\rho \circ \pi)(\Phi, \Psi) = 2\big(\Phi - \Psi A^T,\ \Psi - \Phi A\big). \tag{4.15}$$

Proof. See SM4. ∎

We provide Algorithm 4.1, below, to compute the gradient according to Theorem 4.2, with the appropriate terms given in Proposition 4.3. In SM4.1, we provide Algorithm SM4.1 for computing the gradient of an objective where the modeling error is integrated over time rather than being summed over $\{t_l\}_{l=0}^{L-1}$.

The computational cost of Algorithm 4.1 is dominated by the steps that require evaluation of objects resembling the full-order dynamics, namely, steps 2 and 7. These are of three kinds: evaluating the nonlinear right-hand side $f(x, u)$, acting on a vector with the linearized right-hand side $\partial f(x, u)/\partial x$, or acting with its transpose $(\partial f(x, u)/\partial x)^T$. For a quadratically bilinear FOM and an r-dimensional reduced-order model, assembling the reduced-order model (step 2) requires $O(r^2)$ FOM-like evaluations. Evaluating $S(t)^*\lambda(t)$ in 7 using (4.11) involves querying $f(x, u)$ and $(\partial f(x, u)/\partial x)^T$ acting on a vector. Hence, the cost (per iteration) of step 7 is $O(q)$ FOM-like evaluations, where q is the number of quadrature points used to approximate the integral over the interval $[t_l, t_{l+1}]$. When using high-order quadrature, one may take q to be between one and ten. Thus, the total cost to compute the gradient is $O(r^2 + qL)$ FOM-like evaluations. Most (if not all) modern fluid flow solvers are equipped with the necessary functionality to perform all the aforementioned FOM-like evaluations, so the method that we propose can be easily integrated with existing software.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Algorithm 4.1 Compute the cost function gradient with respect to (Φ, Ψ)

Input: Orthonormal representatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$ with $\det(\Psi^T \Phi) > 0$, initial condition x_0, observations $\{y_l\}_{l=0}^{L-1}$, regularization weight γ.
Assemble and simulate the reduced-order model representative (3.3) from initial condition $z_0 = \Psi^T x_0$, storing predicted outputs $\{\hat{y}_l\}_{l=0}^{L-1}$ and trajectory $z(t)$ via interpolation.
Initialize the gradient: $\operatorname{grad} \bar{J} \leftarrow T(t_{L-1})^* \operatorname{grad} L_y(\hat{y}_{L-1} - y_{L-1})$.
Compute adjoint variable at final time: $\lambda(t_{L-1}) = H(t_{L-1})^* \operatorname{grad} L_y(\hat{y}_{L-1} - y_{L-1})$.
for $l = L-2, L-3, \ldots, 0$ **do**
    Solve the adjoint equation (4.8a) backward in time over the interval $[t_l, t_{l+1}]$ using the linearized reduced-order model dynamics (4.10), and store $\lambda(t)$ on this interval.
    Compute the integral component of (4.9) over the interval $[t_l, t_{l+1}]$: $\operatorname{grad} \bar{J} \leftarrow \operatorname{grad} \bar{J} + \int_{t_l}^{t_{l+1}} S(t)^* \lambda(t) \, \mathrm{d}\, t$ using Gauss–Legendre quadrature.
    Add lth element of the sum in (4.9): $\operatorname{grad} \bar{J} \leftarrow \operatorname{grad} \bar{J} + T(t_l)^* \operatorname{grad} L_y(\hat{y}_l - y_l)$.
    Add "jump" (4.8b) to adjoint variable: $\lambda(t_l) \leftarrow \lambda(t_l) + H(t_l)^* \operatorname{grad} L_y(\hat{y}_l - y_l)$.
end for
Add gradient due to initial condition: $\operatorname{grad} \bar{J} \leftarrow \operatorname{grad} \bar{J} + (\frac{\partial z}{\partial(\Phi,\Psi)}(t_0))^* \lambda(t_0)$.
Normalize by trajectory length: $\operatorname{grad} \bar{J} \leftarrow \operatorname{grad} \bar{J} / L$.
Add regularization: $\operatorname{grad} \bar{J} \leftarrow \operatorname{grad} \bar{J} + \gamma \operatorname{grad}(\rho \circ \pi)(\Phi, \Psi)$.
return $\operatorname{grad} \bar{J}$

Remark 4.4. For a general nonlinear system with sparse coupling between states the discrete empirical interpolation method (DEIM) [16] could eliminate the costly reduced-order model assembly step by replacing f with Πf in (2.4), where Π is the DEIM projector.

4.2. Geometric conjugate gradient algorithm. In Algorithm 4.2, below, we give the implementation details for the geometric conjugate gradient method described by Sato [39], with the required retraction and vector transport provided by the exponential map and parallel translation along geodesics described by Theorems 2.3 and 2.4 in Edelman, Arias, and Smith [18]. Given a search direction $\eta_k = (\xi_k, \zeta_k) \in T_{p_k}(\mathcal{G}_{n,r} \times \mathcal{G}_{n,r})$ at the current iterate $p_k = (V_k, W_k)$ and a step size $\alpha_k \geq 0$, the next iterate is computed using the exponential map [17, 18] according to

$$p_{k+1} = \exp_{p_k}(\alpha_k \eta_k) = \left(\exp_{V_k}(\alpha_k \xi_k),\ \exp_{W_k}(\alpha_k \zeta_k)\right). \tag{4.16}$$

Confining our attention to the one-dimensional objective $J_k(\alpha) = J(\exp_{p_k}(\alpha \eta_k))$ defined along the resulting geodesic, the step size α_k is selected in order to satisfy the Wolfe conditions

$$J_k(\alpha_k) \leq J_k(0) + c_1 \alpha_k J_k'(0), \tag{4.17a}$$

$$J_k'(\alpha_k) \geq c_2 J_k'(0), \tag{4.17b}$$

where $0 < c_1 < c_2 < 1$ are user-specified constants and J_k' denotes the derivative of J_k. Such a step size can always be found, and we use the simple bisection method described in [15] to find one.

The search direction incorporates second-order information about the cost function by combining the gradient of the cost at the current iterate with the previous search direction. The previous search direction $\eta_{k-1} \in T_{p_{k-1}}\mathcal{M}$ is moved to the tangent space at the current iterate $T_{p_k}\mathcal{M}$ via parallel translation [17] along the geodesic, denoted $\mathcal{T}_{p,\eta} : T_p\mathcal{M} \to T_{R_p(\eta)}\mathcal{M}$. We use the explicit formula for parallel

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 4.2 Geometric conjugate gradient algorithm for model reduction

Input: Orthonormal representatives (Φ_0, Ψ_0) of initial subspaces with $\det(\Psi_0^T \Phi_0) > 0$, stopping threshold $\varepsilon > 0$, and Wolfe condition coefficients $0 < c_1 < c_2 < 1$.
Compute cost $\bar{J}(\Phi_0, \Psi_0)$ and gradient $\operatorname{grad} \bar{J}_0$ using Algorithm 4.1
Initialize the search direction $(X_0, Y_0) = \operatorname{grad} \bar{J}_0$, and set $k = 0$.
while $\left\langle \operatorname{grad} \bar{J}_k, \operatorname{grad} \bar{J}_k \right\rangle_{(\Phi_k, \Psi_k)} > \varepsilon$, given by (4.1), **do**
  Compute SVDs $X_k = U_X \Sigma_X V_X^T$, $Y_k = U_Y \Sigma_Y V_Y^T$ with $VV^T = I_r$.
  Define geodesic curves (via [18, Thm. 2.3])

$$\Phi(\alpha) = \left[\Phi_k V_X \cos(\alpha \Sigma_X) + U_X \sin(\alpha \Sigma_X)\right] V_X^T,$$

$$\Psi(\alpha) = \left[\Psi_k V_Y \cos(\alpha \Sigma_Y) + U_Y \sin(\alpha \Sigma_Y)\right] V_Y^T$$

  and line-search objective $J_k(\alpha) = \bar{J}(\Phi(\alpha), \Psi(\alpha))$.
  Compute step size α_k satisfying Wolfe conditions (4.17) using bisection [15] and orthonormal representatives of next iterate $(\Phi_{k+1}, \Psi_{k+1}) = (\Phi(\alpha_k), \Psi(\alpha_k))$.
  Compute parallel translation of search direction (via [18, Thm. 2.4])

$$\tilde{X}_k = \left[-\Phi_k V_X \sin(\alpha_k \Sigma_X) + U_X \cos(\alpha_k \Sigma_X)\right] U_X^T X_k + X_k - U_X U_X^T X_k,$$

$$\tilde{Y}_k = \left[-\Psi_k V_Y \sin(\alpha_k \Sigma_Y) + U_Y \cos(\alpha_k \Sigma_Y)\right] U_Y^T Y_k + Y_k - U_Y U_Y^T Y_k.$$

  Multiply first column of Φ_{k+1} and $\tilde{X}_k$ by $\operatorname{sgn} \det(\Psi_{k+1}^T \Phi_{k+1})$.
  Compute cost $\bar{J}(\Phi_{k+1}, \Psi_{k+1})$ and gradient $\operatorname{grad} \bar{J}_{k+1}$ using Algorithm 4.1.
  Using (4.1), compute Riemannian Dai–Yuan coefficient

$$\beta_{k+1} = \frac{\langle \operatorname{grad} \bar{J}_{k+1}, \operatorname{grad} \bar{J}_{k+1} \rangle_{(\Phi_{k+1}, \Psi_{k+1})}}{\langle \operatorname{grad} \bar{J}_{k+1}, (\tilde{X}_k, \tilde{Y}_k) \rangle_{(\Phi_{k+1}, \Psi_{k+1})} + \langle \operatorname{grad} \bar{J}_k, (X_k, Y_k) \rangle_{(\Phi_k, \Psi_k)}}.$$

  Compute next search direction $(X_{k+1}, Y_{k+1}) = \operatorname{grad} \bar{J}_{k+1} + \beta_{k+1} (\tilde{X}_k, \tilde{Y}_k)$.
  Update $k \leftarrow k + 1$.
end while
return orthonormal representatives (Φ_K, Ψ_K) of the optimized projection subspaces and the final cost $\bar{J}(\Phi_K, \Psi_K)$

translation along geodesics on the Grassmann manifold given by Theorem 2.4 in [18] and the fact that $\mathcal{T}_{(V,W),(\xi_1,\zeta_1)}(\xi_2, \zeta_2) = (\mathcal{T}_{V,\xi_1} \xi_2,\ \mathcal{T}_{W,\zeta_1} \zeta_2)$ on the product manifold $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. The next search direction is computed according to

$$\eta_k = -\operatorname{grad} J(p_k) + \beta_k \mathcal{T}_{p_{k-1}, \alpha_{k-1} \eta_{k-1}}(\eta_{k-1}), \tag{4.18}$$

where the coefficient β_k is defined differently for different conjugate gradient algorithms. We use the Riemannian Dai–Yuan coefficient proposed by Sato [39] and given by

$$\beta_k = \frac{\langle \operatorname{grad} J(p_k),\ \operatorname{grad} J(p_k) \rangle_{p_k}}{\left\langle \operatorname{grad} J(p_k),\ \mathcal{T}_{\alpha_{k-1} \eta_{k-1}}(\eta_{k-1}) \right\rangle_{p_k} - \langle \operatorname{grad} J(p_{k-1}),\ \eta_{k-1} \rangle_{p_{k-1}}} \tag{4.19}$$

since it yields excellent performance in practice and provides guaranteed convergence when the step sizes satisfy the Wolfe conditions [48].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

4.3. Convergence guarantees. The proofs of convergence for the geometric conjugate gradient algorithms described in [36, 40, 39] (with retraction provided by the exponential map) rely on Lipschitz assumptions for the derivative of $J \circ \exp_{p_k}$ along the search direction η_k. In particular, if there is a fixed Lipschitz constant L_J so that for each iteration $k = 0, 1, 2, \ldots$, we have

$$\left|D(J \circ \exp_{p_k})(\alpha_k \eta_k)\eta_k - D(J \circ \exp_{p_k})(0)\eta_k\right| \leq L_J \alpha_k \|\eta_k\|_{p_k}^2, \tag{4.20}$$

then the Riemannian generalization of Zoutendijk's theorem given by Theorem 2 in [36] (Theorem 4.1 in [39]) holds, and Theorem 4.2 in [39] guarantees convergence of the geometric conjugate gradient algorithm in the sense that

$$\boxed{\liminf_{k\to\infty} \left\| \operatorname{grad} J(V_k, W_k)\right\|_{(V_k,W_k)} = 0.} \tag{4.21}$$

In other words, the conjugate gradient algorithm will produce iterates with arbitrarily small gradients, which may be used as a stopping condition. Fortunately, the Lipschitz condition (4.20) is easily verified, and we obtain the following convergence result.

THEOREM 4.5. *Suppose that there is a compact subset $\mathcal{D}_c$ of the domain $\mathcal{D}$ (defined in Proposition 3.2) such that, for every iteration $k = 0, 1, 2, \ldots$, we have*

$$\gamma_k(t) = \exp_{p_k}(t\eta_k) \in \mathcal{D}_c \qquad \forall t \in [0, \alpha_k]. \tag{4.22}$$

Let ∇ denote the Riemannian connection on $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ with a metric given by (4.4). *Then the Lipschitz condition* (4.20) *holds with*

$$L_J = \max_{\substack{(p,\xi)\in T\mathcal{M}:\\ p\in\mathcal{D}_c,\ \|\xi\|_p=1}} \left\|(\nabla_\xi \operatorname{grad} J)(p)\right\|_p < \infty, \tag{4.23}$$

and the geometric conjugate gradient algorithm with Dai–Yuan coefficient (4.19) *and α_k satisfying the Wolfe conditions* (4.17) *converges in the sense of* (4.21).

Proof. See SM5. ∎

The Lipschitz estimate in Theorem 4.5 also guarantees the convergence of other geometric conjugate gradient algorithms such as the Riemannian Fletcher–Reeves method with strengthened Wolfe conditions presented in [36].

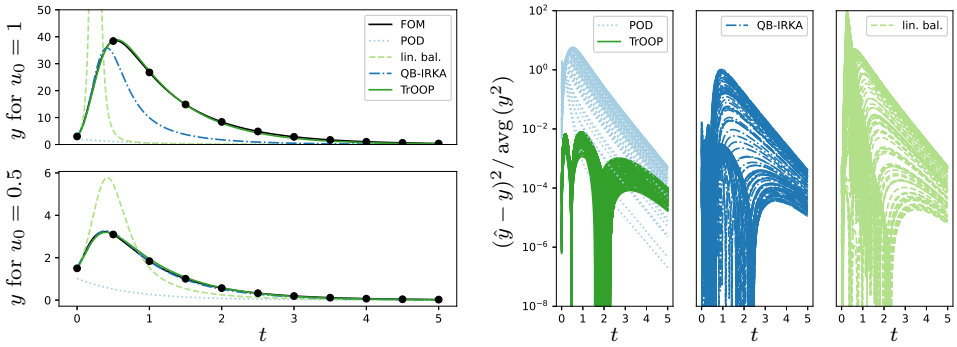
Remark 4.6. As we discuss in SM5, it is always possible to find step sizes α_k that satisfy the Wolfe conditions and the assumption in Theorem 4.5. However, guaranteeing that the step size produces a path contained in a predefined compact set $\mathcal{D}_c$ requires modifying the line-search method. In practice, we did not find such a modification necessary, and the simple bisection method in [15] was sufficient to produce converging iterates in Algorithm 4.2.

5. Simple nonlinear system with an important low-energy feature. In this section, we illustrate our method on a simple example system for which existing approaches to nonlinear model reduction perform poorly. In particular, we consider the system

$$\begin{aligned} \dot{x}_1 &= -x_1 + 20x_1x_3 + u, \\ \dot{x}_2 &= -2x_2 + 20x_2x_3 + u, \\ \dot{x}_3 &= -5x_3 + u, \\ y &= x_1 + x_2 + x_3, \end{aligned} \tag{5.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) training trajectories (b) error on testing trajectories

FIG. 1. *In panel* (a)*, we show the outputs generated by the FOM* (5.1) *and various two-dimensional projection-based reduced-order models in response to impulses with magnitudes* $u_0 = 0.5$ *and* $u_0 = 1$ *at* $t = 0$*. The sample points used to construct the objective function* (5.2) *used to optimize the projection operator are shown as black dots. In panel* (b)*, we show the normalized square errors of the reduced-order model predictions in response to* 100 *impulses at* $t = 0$ *whose magnitudes* u_0 *were drawn uniformly at random from the interval* $[0, 1]$.

and we compare our method (TrOOP) with two-dimensional projection-based models obtained using subspaces determined by POD, balanced truncation of the linearized system, quadratic-bilinear (QB) balanced truncation [10], and the QB IRKA (QB-IRKA) presented in [11]. The QB-balancing method had similar but slightly worse performance than QB-IRKA, so we shall only show the results using QB-IRKA. We confine our attention to nonlinear impulse responses with magnitudes $u_0 \in [0, 1]$. These responses can be obtained by considering the output of (5.1) with $u \equiv 0$ and known initial condition $x(0) = u_0(1,\ 1,\ 1)$. Two such responses with $u_0 = 0.5$ and $u_0 = 1$ are shown in Figure 1(a).

The key feature of (5.1) is that the state x_3 plays a very important role in the dynamics of the states x_1 and x_2 while remaining small by comparison due to its fast decay rate. In fact, for $u_0 > 1/5$ we have $\dot{y}(0) > 0$, and the output experiences transient growth due to the nonlinear interaction of x_1 and x_2 with x_3. These nonlinear interactions become dominant for larger u_0 but are neglected completely by model reduction techniques like balanced truncation that consider only the linear part of (5.1). Figure 1(a) shows the result of such an approach, in which we obtain a nonlinear reduced-order model by Petrov–Galerkin projection of (5.1) onto a two-dimensional subspace determined by balanced truncation of the linearized system. As shown in the figure, the resulting model overpredicts the transient growth by an amount that increases with u_0. Techniques such as QB-balancing and QB-IRKA extend the region of validity for the reduced-order models by considering second-order terms in the Volterra series for the response, yet still have deteriorating accuracy with increasing u_0 due to the neglected higher-order terms.

On the other hand, a two-dimensional POD-based model retains the most energetic states, which align closely with x_1 and x_2, and essentially ignores the important low-energy state x_3. Consequently, the POD-based model of (5.1) does not predict any transient growth as shown in Figure 1(a).

In order to find a two-dimensional reduced-order model of (5.1) using TrOOP, we collected the two impulse-response trajectories shown in Figure 1(a) and used the $L = 11$ equally spaced samples shown for each trajectory to define the cost function

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$$J(V,W) = \sum_{u_0 \in \{0.5,1.0\}} \frac{1}{\sum_{l=0}^{L-1} (y|_{u_0}(t_l))^2} \sum_{l=0}^{L-1} \left(\hat{y}|_{u_0}(t_l) - y|_{u_0}(t_l)\right)^2 + \gamma\rho(V,W), \tag{5.2}$$

with $\gamma = 10^{-3}$ (although we note that the results were not sensitive to the choice of γ). The normalizing factor in the cost for each trajectory was used to penalize the error relative to the average energy content of the trajectory rather than in an absolute sense which would be dominated by the trajectory with $u_0 = 1$. Starting from an initial model formed by balanced truncation, the conjugate gradient algorithm described above with Wolfe conditions defined by $c_1 = 0.01$ and $c_2 = 0.1$ achieves convergence with a gradient magnitude smaller than 10^{-4} after fewer than 150 steps (depending on the ODE solver). We note that initialization using POD fails to produce an accurate model because the optimization converges to a local minimum of (5.2).

In Figure 1(a), we see that the resulting reduced-order model trajectories very closely match the trajectories used to find the oblique projection. Moreover, we tested the predictions of the reduced-order models on 100 impulse-response trajectories with u_0 drawn uniformly at random from the interval $[0, 1]$. The square output prediction errors for each trajectory normalized by the average output energy of the FOM are shown in Figure 1(b). We observe that the POD-based model is poor regardless of the impulse magnitude u_0, whereas Petrov–Galerkin projection onto subspaces determined by linear balanced truncation performs well when u_0 is very close to 0 but poorly when u_0 is closer to 1. The projection subspaces obtained by QB-IRKA (and QB-balancing) yield models that are accurate in a larger neighborhood of the origin than balanced truncation of the linearized dynamics, yet still perform poorly for large u_0. On the other hand, the reduced-order model we found using TrOOP produces very accurate predictions for all impulse-response magnitudes in the desired range. This model also has excellent predictive performance with different input signals, even though it was optimized using only two impulse responses. For instance, Figure 2 shows the predictions of the reduced-order models in response to a sinusoidal input $u(t) = \sin(t)$ with zero initial condition.

6. Reduction of a high-dimensional nonlinear fluid flow. In this section we set out to develop reduced-order models capable of predicting the response of an incompressible jet flow to disturbances in the proximity of the nozzle. We consider the evolution of an axisymmetric jet flow over the spatial domain $\Omega = \{(\xi, z) \mid \xi \in [0, L_\xi], z \in [0, L_z]\}$. Here, ξ denotes the radial direction, and z denotes the axial direction. Velocities are nondimensionalized by the centerline velocity U_0, lengths by the jet diameter D_0, and pressure by ρU_0^2, where ρ is the fluid density. Letting $q = (u, v)$ denote the (dimensionless) velocity vector with axial component u and radial component v and letting p be the (dimensionless) pressure field, we may write the governing equations in cylindrical coordinates as

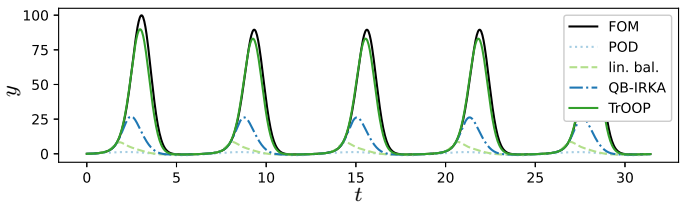


FIG. 2. *We show the responses of* (5.1) *and the reduced-order models to input* $u(t) = \sin(t)$.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial z} - v\frac{\partial u}{\partial \xi} - \frac{\partial p}{\partial z} + \frac{1}{Re}\left(\frac{1}{\xi}\frac{\partial}{\partial \xi}\left(\xi\frac{\partial u}{\partial \xi}\right) + \frac{\partial^2 u}{\partial z^2}\right), \tag{6.1}$$

$$\frac{\partial v}{\partial t} = -u\frac{\partial v}{\partial z} - v\frac{\partial v}{\partial \xi} - \frac{\partial p}{\partial \xi} + \frac{1}{Re}\left(\frac{1}{\xi}\frac{\partial}{\partial \xi}\left(\xi\frac{\partial v}{\partial \xi}\right) - \frac{v}{\xi^2} + \frac{\partial^2 v}{\partial z^2}\right), \tag{6.2}$$

$$\frac{\partial u}{\partial z} + \frac{1}{\xi}\frac{\partial}{\partial \xi}(\xi v) = 0, \tag{6.3}$$

where $Re = U_0 D_0/\nu$ is the Reynolds number and ν is the kinematic viscosity of the fluid. Throughout, we take $Re = 1000$. The algebraic constraint in formula (6.3) may be used to eliminate pressure from formulas (6.1) and (6.2), as discussed in SM6. We impose zero gradient boundary conditions on the velocity at the top boundary $\xi = L_\xi$ and at the outflow boundary $z = L_z$, and we let the inflow velocity be

$$u(\xi, 0) = \frac{1}{2}\left(1 - \tanh\left[\frac{1}{4\theta_0}\left(\xi - \frac{1}{\xi}\right)\right]\right), \tag{6.4}$$

where θ_0 is a dimensionless thickness, which we fix at $\theta_0 = 0.025$. The equations of motion are integrated in time using the fractional step method described in [34] in conjunction with the second-order Adams–Bashforth multistep scheme. The spatial discretization is performed on a fully staggered grid of size $N_z \times N_\xi = 250 \times 200$ and with $L_z = 10$ and $L_\xi = 4$. If we let the state be composed of the axial and radial velocities at the cell faces, then the state dimension for this flow is $2(N_z \times N_\xi) = 10^5$. The solver has been validated against some of the results presented in [44], for which we observed very good quantitative agreement. Throughout this section, the inner product on the state space is given by

$$\langle f, g\rangle = \int_\Omega f(\xi, z)^T g(\xi, z)\, \xi \, \mathrm{d}\, \xi \, \mathrm{d}\, z. \tag{6.5}$$

This may be transformed into a Euclidean inner product by scaling the elements of the state space by $\sqrt{\xi}$. We take our observations, y, to be the full velocity field on the spatial grid scaled by $\sqrt{\xi}$.

6.1. Results. For the described flow configuration, there exists a stable steady-state solution, which we will denote Q. Any perturbation q' about the steady-state solution will grow while advecting downstream, and it will eventually leave the computational domain through the outflow located at $z = L_z$. During the growth process, nonlinear effects become dominant and lead to the formation of complicated vortical structures. In this section we seek to develop a reduced-order model of the growth of these disturbances in response to impulses that enter the radial momentum equation (6.2) through a velocity perturbation localized near $\xi = 1/2$ and $z = 1$. In particular, the perturbation has the form $B(\xi, z)w(t)$, where

$$B(\xi, z) = \exp\left\{-\frac{(\xi - 1/2)^2 + (z-1)^2}{\theta_0}\right\}. \tag{6.6}$$

We simulate the response of the flow to a given impulse $w(t) = \alpha\delta(t)$, with $\alpha \in \mathbb{R}$, by integrating the governing equations (6.1)–(6.3) with initial condition

$$q(0) = Q + q'(0), \quad \text{where} \quad q'(0) = (0, B\alpha). \tag{6.7}$$

Here we construct reduced-order models to capture the response of the flow to impulses with $-1.0 \le \alpha \le 1.0$ from the initial time $t = 0$ to a final time ($t \approx 30$) when

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

all disturbances have left the computational domain through the outflow boundary located at $z = L_z$.

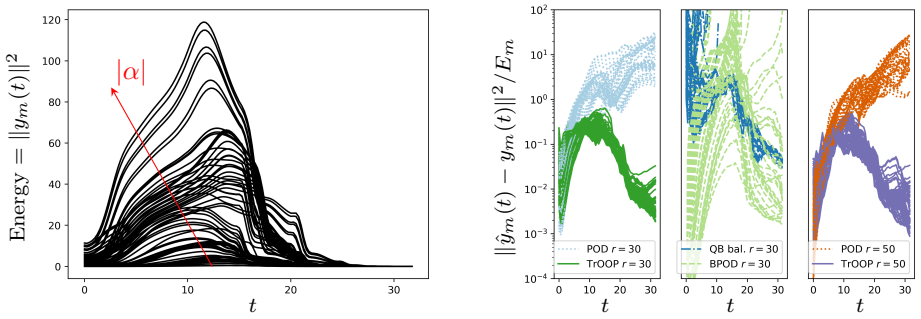
We proceed as follows: we generate a training set of $M = 14$ trajectories corresponding to values $\alpha \in \pm\{0.01, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$, and from each trajectory we observe $L = 64$ equally spaced snapshots of velocity perturbations about the base flow Q. Let $y_{m,l}$ denote the lth velocity snapshot in the mth trajectory, and let $\hat{y}_{m,l}$ denote the corresponding prediction obtained by integrating the reduced-order model from the initial condition $\hat{q}'_{m,0} = P_{V,W} q'_{m,0}$. Letting $E_m = L^{-1} \sum_{l=0}^{L-1} \|y_{m,l}\|^2$ denote the average energy along the mth trajectory, we seek to minimize the cost function

$$J(V, W) = \frac{1}{ML} \sum_{m=0}^{M-1} \frac{1}{E_m} \sum_{l=0}^{L-1} \|\hat{y}_{m,l} - y_{m,l}\|^2 + \gamma \rho(V, W), \tag{6.8}$$

where $\gamma = 10^{-3}$. The optimization was carried out using Algorithm 4.2 with an r-dimensional model obtained by POD of all available training snapshots as the initial guess. As we will see below, models obtained using BPOD and QB-balancing experienced blowup, which prevented us from using these methods to initialize TrOOP. The integrals in Algorithm 4.1 were computed using Gauss–Legendre quadrature with two quadrature points between adjacent FOM data points. Here, we train two models: one with $r = 30$ and one with $r = 50$, where the first 30 POD modes accounted for 98.6% of the training energy and the first 50 accounted for 99.6% of the energy.

We compare the models obtained using TrOOP to projection-based models of the same dimension using subspaces determined by POD, BPOD, and QB-balancing on a set of $M = 65$ unseen impulse responses. For 50 of these, α was drawn uniformly at random from $[-1.0, 1.0]$, while the remaining 15 were drawn uniformly from $[-0.1, 0.1]$. The energy content of the testing set is shown in Figure 3(a). Observe the range of behavior for different values of α, reflecting the strong nonlinearity of this flow. The BPOD model was obtained following the procedure discussed in [37], with a 30-dimensional output projection that accounted for more than 99.9% of the energy. Our implementation of the QB-balancing method is discussed in SM7.

The performance of each model is shown in Figure 3(b), with the exception of BPOD and QB-balancing with $r = 50$. The predictions from these models blew up



(a) testing data (b) error on testing trajectories

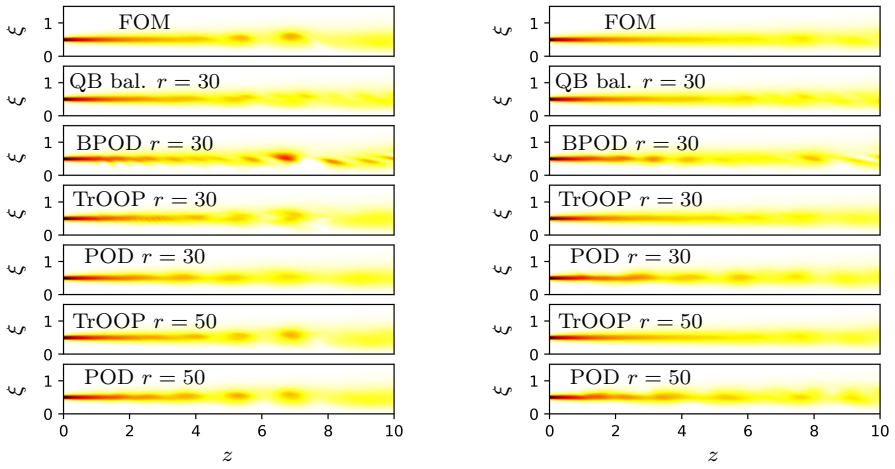
FIG. 3. *In panel* (a) *we show the time history of the energy of the impulse responses in the training data set. In panel* (b) *we show the square error across all training trajectories for the optimal reduced-order model and for the POD-based model with dimensions* $r = 30$ *and* $r = 50$ *and for the BPOD-based model and for the QB-balancing model of dimension* $r = 30$.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

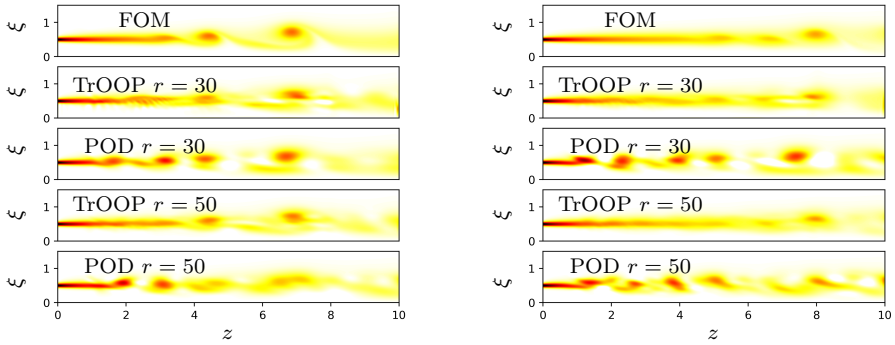
for virtually all amplitudes α. Figure 3(b) shows that the POD/Galerkin models accurately represent the initial growth of the perturbations at all amplitudes, but they perform poorly at long times. The QB-balancing model exhibits large errors at initial times, and it blows up for many of the testing trajectories with larger values of α. The BPOD-based model performs very well for small amplitudes α, but it too performs poorly or even blows up for larger values of α. By contrast, the models obtained using TrOOP are accurate over the entire time-horizon at every amplitude and capture the initial transient growth of the perturbations as well as the long-time decay.

Snapshots extracted from the trajectories with $\alpha = 0.158$ and $\alpha = -0.943$ are, respectively, shown in Figure 4 and Figure 5. Results are not shown for BPOD and QB-balancing in Figure 5 because these models blew up after a few time steps at the higher amplitude. At both amplitudes the optimized models correctly predict the



(a) vorticity at time $t = 12$ and $\alpha = 0.158$ (b) vorticity at time $t = 18$ and $\alpha = 0.158$

FIG. 4. *In panel* (a) *we show a vorticity snapshot (i.e.,* $\nabla \times (q' + Q)$*) at time* $t = 12$ *from the trajectory generated by the impulse with* $\alpha = 0.158$. *In panel* (b) *we show the analogue of panel* (a) *with* $t = 18$ *and* $\alpha = 0.158$. *The color bar ranges from* 0 *(white) to approximately* 9 *(red).*

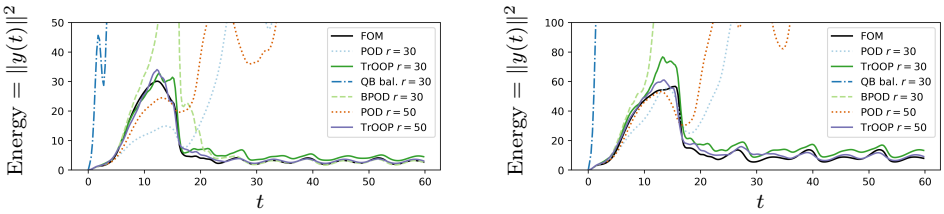


(a) vorticity at time $t = 12$ and $\alpha = -0.943$ (b) vorticity at time $t = 18$ and $\alpha = -0.943$

FIG. 5. *Analogue of Figure* 4 *except with* $\alpha = -0.943$. *The BPOD and QB-balancing predictions are not shown because they blew up after a few time steps.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) forcing with $-0.3B(r,z)\cos(t/2)$ (b) forcing with $-0.5B(r,z)\cos(t/2)$

FIG. 6. *In panel* (a) *we show the energy of the response of the flow to a radial velocity input* $w(r,z,t) = -\beta B(r,z)\cos(t/2)$ *with* $\beta = -0.3$. *Panel* (b) *is the analogue of panel* (a) *with* $\beta = -0.5$.

location and strength of the vortical structures that form in response to the initial impulse, while the other reduced-order models exhibit spurious vortical structures or begin to lose predictive accuracy at long times.

The models found using TrOOP are also able to predict the response of the flow to other types of input signals. For example, we consider a radial velocity input of the form $w(r,z,t) = -\beta B(r,z)\cos(t/2)$ for values $\beta = 0.3$ and $\beta = 0.5$. The results are shown in Figure 6(a) and Figure 6(b), where we plot the predicted energy of the velocity field over time. For both amplitudes, our models correctly capture the qualitative nature of the response of the flow, and the 50-dimensional model also exhibits very good quantitative agreement. By contrast, the QB-balancing model "blows up" at early times, both POD/Galerkin models blow up at later times, and the BPOD model either exhibits extremely large transient growth at $\beta = 0.3$ or it blows up for $\beta = 0.5$. It is worth mentioning that the 30-dimensional BPOD model has excellent performance on the low-amplitude posttransient response.

6.2. Computational cost and considerations. Here, we provide a brief comparison of the computational costs of each method for the jet flow in terms of the number of times an object resembling the right-hand side of the FOM is evaluated, i.e., $f(x,u)$, $(\partial f(x,u)/\partial x)v$, or $(\partial f(x,u)/\partial x)^T v$ acting on a single vector $v \in \mathbb{R}^n$. Such evaluations dominate the computational cost of each method we considered. We recall that TrOOP assembles the reduced-order model at each line-search iteration using queries to $f(x,u)$, while the gradient is computed using Algorithm 4.1 by querying $f(x,u)$ and $(\partial f(x,u)/\partial x)^T v$ at quadrature points along each trajectory. Since solving the Lyapunov equations for balanced truncation [33] and its QB extension [10] were infeasible for our system with 10^5 states, we used BPOD [37] and an analogous snapshot-based approximation for QB-balancing described in SM7 involving similar queries. In fact, to the best of our knowledge, the QB-balancing algorithm has never been applied to systems with a state dimension larger than $\sim 10^3$. Table 1 summarizes the total cost of each method, and Figure 7 shows the progress of the conjugate gradient algorithm against the number of iterations and the total number of FOM-like evaluations.

TABLE 1

We compare the number of FOM-like evaluations for each model-reduction technique on the jet flow to a single simulation of the FOM from $t = 0$ *to* $t = 30$ *using the time step* $\Delta t = 5 \times 10^{-3}$ *and the second-order Adams–Bashforth method.*

Method	FOM sim.	BPOD	QB bal.	TrOOP $r = 30$	TrOOP $r = 50$
FOM evals.	6×10^3	2×10^5	1.3×10^6	2.3×10^6	2.1×10^6

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

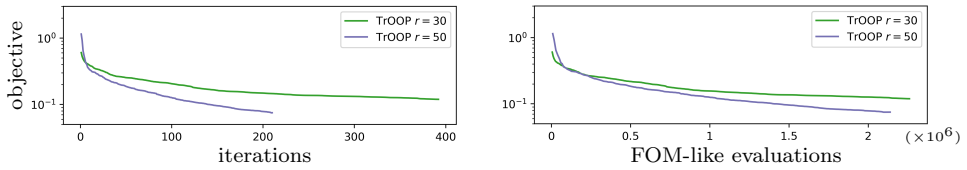


FIG. 7. *Values of the jet flow optimization objective* (6.8) *versus conjugate gradient iterations (left) and FOM-like evaluations (right).*

7. Conclusions. We have introduced a reduced-order modeling approach for large-scale nonlinear dynamical systems based on optimizing oblique projections of the governing equations to minimize prediction error over sampled trajectories. We implemented a provably convergent geometric conjugate gradient algorithm in order to optimize a regularized trajectory prediction error over the product of Grassmann manifolds defining the projection operators. The method, referred to as TrOOP, is compared to existing projection-based reduced-order modeling techniques, where the projection subspaces are found using POD, balanced truncation, and techniques for QB systems. We considered a simple three-dimensional system with an important low-energy feature as well as a nonlinear axisymmetric jet flow with 10^5 state variables. In both cases, the models obtained using TrOOP vastly outperform the models obtained using other methods in the highly nonlinear regimes far away from equilibria while achieving comparable performance to the best alternatives near equilibria. The algorithms were implemented in Python and run on a personal computer. Our code that implements TrOOP is available at https://github.com/samotto1/TrOOP.

There are two key issues that we would like to address in future work. First, the performance of TrOOP depends on the initial subspaces, which may be poor. Second, chaotic dynamics do not admit accurate predictions over long time-horizons. Preliminary experiments with the jet flow at higher Reynolds numbers indicate that it may be helpful to break long trajectories into pieces during training and to increase the length of the pieces as the model improves. It is also necessary to use a sufficiently large collection of trajectories in order to sample the system's behavior and avoid overfitting. Based on algebraic considerations, the total number of sample data should exceed the dimension $2nr - 2r^2$ of the product of Grassmann manifolds over which we optimize, where n is the state dimension and r is the dimension of the reduced-order model. When a large number of trajectories is used, it may be advantageous to employ a stochastic gradient descent algorithm [13, 41] with randomized "minibatches" of trajectories.

REFERENCES

[1] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Riemannian geometry of Grassmann manifolds with a view on algorithmic computation*, Acta Appl. Math., 80 (2004), pp. 199–220.
[2] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization algorithms on matrix manifolds*, Princeton University Press, Princeton, NJ, 2009.
[3] S. AHUJA AND C. W. ROWLEY, *Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators*, J. Fluid Mech., 645 (2010), pp. 447–478.
[4] A. C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.
[5] A. BARBAGALLO, D. SIPP, AND P. J. SCHMID, *Closed-loop control of an open cavity flow using reduced-order models*, J. Fluid Mech., 641 (2009), p. 1–50.
[6] U. BAUR, P. BENNER, AND L. FENG, *Model order reduction for linear and nonlinear systems: A system-theoretic perspective*, Arch. Comput. Methods Eng., 21 (2014), pp. 331–358.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

[7] T. Bendokat, R. Zimmermann, and P.-A. Absil, *A Grassmann Manifold Handbook: Basic Geometry and Computational Aspects*, preprint, arXiv:2011.13699, 2020.
[8] P. Benner and T. Breiten, *Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 859–885.
[9] P. Benner and T. Breiten, *Two-sided projection methods for nonlinear model order reduction*, SIAM J. Sci. Comput., 37 (2015), pp. B239–B260.
[10] P. Benner and P. Goyal, *Balanced Truncation Model Order Reduction for Quadratic-Bilinear Control Systems*, preprint, arXiv:1705.00160, 2017.
[11] P. Benner, P. Goyal, and S. Gugercin, *$\mathcal{H}_2$-quasi-optimal model order reduction for quadratic-bilinear control systems*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 983–1032.
[12] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531.
[13] S. Bonnabel, *Stochastic gradient descent on Riemannian manifolds*, IEEE Trans. Automat. Control, 58 (2013), pp. 2217–2229.
[14] G. L. Brown and A. Roshko, *On density effects and large structure in turbulent mixing layers*, J. Fluid Mech., 64 (1974), pp. 775–816.
[15] J. V. Burke, *Line Search Methods*, lecture notes for MATH 408, University of Washington, 2014, available online at https://sites.math.washington.edu/~burke/crs/408/notes/nlp/line.pdf.
[16] S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764.
[17] M. P. do Carmo, *Riemannian Geometry*, Birkhäuser, Basel, 1992.
[18] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.
[19] G. Flagg and S. Gugercin, *Multipoint Volterra series interpolation and $\mathcal{H}_2$ optimal model reduction of bilinear systems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 549–579.
[20] S. Gugercin, A. C. Antoulas, and C. Beattie, *$\mathcal{H}_2$ model reduction for large-scale linear dynamical systems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638.
[21] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge University Press, Cambridge, UK, 2012.
[22] W. Huang, K. A. Gallivan, and P.-A. Absil, *A Broyden class of quasi-Newton methods for Riemannian optimization*, SIAM J. Optim., 25 (2015), pp. 1660–1685.
[23] M. Ilak, S. Bagheri, L. Brandt, C. W. Rowley, and D. S. Henningson, *Model reduction of the nonlinear complex Ginzburg–Landau equation*, SIAM J. Appl. Dyn. Syst., 9 (2010), pp. 1284–1302.
[24] S. J. Illingworth, A. S. Morgans, and C. W. Rowley, *Feedback control of flow resonances using balanced reduced-order models*, J. Sound Vib., 330 (2011), pp. 1567–1581.
[25] Y.-L. Jiang and K.-L. Xu, *Riemannian modified Polak–Ribière–Polyak conjugate gradient order reduced model by tensor techniques*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 432–463.
[26] W. G. Kelly and A. C. Peterson, *The Theory of Differential Equations, Classical and Qualitative*, Prentice-Hall, Englewood Cliffs, NJ, 2004.
[27] B. Kramer and K. E. Willcox, *Balanced Truncation Model Reduction for Lifted Nonlinear Systems*, preprint, arXiv:1907.12084, 2019.
[28] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed., Springer, New York, 2013.
[29] K. Lee and K. T. Carlberg, *Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders*, J. Comput. Phys., 404 (2020), 108973.
[30] J. L. Lumley, *The structure of inhomogeneous turbulent flows*, in Atmospheric Turbulence and Radio Wave Propagation, Nauka, Moscow, 1967, pp. 166–178.
[31] M. Marion and R. Temam, *Nonlinear Galerkin methods*, SIAM J. Numer. Anal., 26 (1989), pp. 1139–1157.
[32] C. D. Meyer, *Matrix analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
[33] B. Moore, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Trans. Automat. Control, 26 (1981), pp. 17–32.
[34] J. B. Perot, *An analysis of the fractional step method*, J. Comput. Phys., 108 (1993), pp. 51–58.
[35] G. Rega and H. Troger, *Dimension reduction of dynamical systems: Methods, models, applications*, Nonlinear Dyn., 41 (2005), pp. 1–15.
[36] W. Ring and B. Wirth, *Optimization methods on Riemannian manifolds and their application to shape space*, SIAM J. Optim., 22 (2012), pp. 596–627.
[37] C. W. Rowley, *Model reduction for fluids, using balanced proper orthogonal decomposition*, Internat. J. Bifurcation Chaos Appl. Sci Engrg., 15 (2005), pp. 997–1013.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[38] C. W. Rowley and S. T. Dawson, *Model reduction for flow analysis and control*, Ann. Rev. Fluid Mech., 49 (2017), pp. 387–417.
[39] H. Sato, *A Dai–Yuan-type Riemannian conjugate gradient method with the weak Wolfe conditions*, Comput. Optim. Appl., 64 (2016), pp. 101–118.
[40] H. Sato and T. Iwai, *A new, globally convergent Riemannian conjugate gradient method*, Optimization, 64 (2015), pp. 1011–1031.
[41] H. Sato, H. Kasai, and B. Mishra, *Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport*, SIAM J. Optim., 29 (2019), pp. 1444–1472.
[42] H. Sato and K. Sato, *Riemannian trust-region methods for $\mathcal{H}_2$ optimal model reduction*, in Proceedings of the 2015 54th IEEE Conference on Decision and Control (CDC), IEEE, 2015, pp. 4648–4655.
[43] P. J. Schmid and D. S. Henningson, *Stability and Transition in Shear Flows*, Appl. Math. Sci. 142, Springer, New York, 2001.
[44] L. Shaabani-Ardali, D. Sipp, and L. Lesshafft, *Vortex pairing in jets as a global Floquet instability: Modal and transient dynamics*, J. Fluid Mech., 862 (2019), pp. 951–989.
[45] L. Sirovich, *Turbulence and the dynamics of coherent structures: Part* I*: Coherent structures*, Quart. Appl. Math., 45 (1987), pp. 561–571.
[46] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll, *Hydrodynamic stability without eigenvalues*, Science, 261 (1993), pp. 578–584.
[47] W.-G. Wang and Y.-L. Jiang, *$\mathcal{H}_2$ optimal model order reduction on the Stiefel manifold for the MIMO discrete system by the cross Gramian*, Math. Comput. Model. Dyn. Syst., 24 (2018), pp. 610–625.
[48] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226–235.
[49] K.-L. Xu and Y.-L. Jiang, *An unconstrained $\mathcal{H}_2$ model order reduction optimisation algorithm based on the Stiefel manifold for bilinear systems*, Internat. J. Control, 92 (2019), pp. 950–959.
[50] Y. Xu and T. Zeng, *Fast optimal $\mathcal{H}_2$ model reduction algorithms based on Grassmann manifold optimization*, Int. J. Numer. Anal. Model., 10 (2013), pp. 972–991.
[51] W.-Y. Yan and J. Lam, *An approximate approach to $\mathcal{H}_2$ optimal model reduction*, IEEE Trans. Automat. Control, 44 (1999), pp. 1341–1358.
[52] P. Yang, Y.-L. Jiang, and K.-L. Xu, *A trust-region method for $\mathcal{H}_2$ model reduction of bilinear systems on the Stiefel manifold*, J. Franklin Inst., 356 (2019), pp. 2258–2273.
[53] T. Zeng and C. Lu, *Two-sided Grassmann manifold algorithm for optimal model reduction*, Internat J. Numer. Meth. Engrg., 104 (2015), pp. 928–943.

Downloaded 11/30/25 to 188.92.136.187 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.