GEOMETRIC SUBSPACE UPDATES WITH APPLICATIONS TO ONLINE ADAPTIVE NONLINEAR MODEL REDUCTION*

RALF ZIMMERMANN†, BENJAMIN PEHERSTORFER‡, AND KAREN WILLCOX§

Abstract. In many scientific applications, including model reduction and image processing, subspaces are used as ansatz spaces for the low-dimensional approximation and reconstruction of the state vectors of interest. We introduce a procedure for adapting an existing subspace based on information from the least-squares problem that underlies the approximation problem of interest such that the associated least-squares residual vanishes exactly. The method builds on a Riemannian optimization procedure on the Grassmann manifold of low-dimensional subspaces, namely the Grassmannian Rank-One Subspace Estimation (GROUSE). We establish for GROUSE a closed-form expression for the residual function along the geodesic descent direction. Specific applications of subspace adaptation are discussed in the context of image processing and model reduction of nonlinear partial differential equation systems.

Key words. online adaptive model reduction; dimension reduction; Grassmann manifold; Grassmannian Rank-One Subspace Estimation (GROUSE); discrete empirical interpolation method (DEIM); gappy proper orthogonal decomposition (POD); masked projection; rank-one updates; optimization on manifolds; subspace fitting; least-squares; image processing

AMS subject classifications. 65F30; 65F20; 49M15; 65K10; 37M99;

1. Introduction.

Dimension reduction techniques play an important role in the application of computational methods—identifying inherent low-dimensional structure in the problem at hand can often lead to significant reductions in computational complexity. Consider a set of state vectors embedded in the n-dimensional Euclidean space $\mathbb{R}^n, n \in \mathbb{N}$. The goal of dimension reduction is to restrict the space of state vector candidates to a subspace of $\mathbb{R}^n$ of low dimension $p \ll n$. In doing so, the n-degree-of-freedom problem of computing full-scale state vectors is replaced by the task of determining the p coefficients of a basis expansion in the reduced subspace. If, for example, the state vectors are solutions of a computational model, then this dimension reduction underlies the derivation of a projection-based reduced model. As another example, the state vectors might represent experimental data or other system samples such as representations of an image. In those cases, the dimension reduction seeks an efficient compression of the data and a low-dimensional subspace in which to reconstruct unknown states. When n is large, dimension reduction often leads to a tremendous reduction in computational complexity; however, acceptable accuracy is only retained if the full state vectors can be approximated well in the p-dimensional subspace. Thus, the identification and numerical representation of subspaces plays a critical role.

In classical projection-based model reduction, the reduced subspace is determined once in a so-called offline phase. Subsequently, it stays fixed while the reduced model is evaluated during the so-called online phase. Online adaptive model reduction breaks

*The first author was funded via a research fellowship awarded by the German Research Foundation (DFG), grant number Zi 1250/2-1. The second and third authors acknowledge support of the U.S. Department of Energy, Awards DE-FG02-08ER2585 and DE-SC0009297, as part of the DiaMonD Multifaceted Mathematics Integrated Capability Center.

†Department of Mathematics and Computer Science, SDU Odense, Denmark and Department of Aeronautics & Astronautics, MIT, Cambridge, USA (zimmermann@imada.sdu.dk).

‡Department of Mechanical Engineering, University of Wisconsin-Madison, USA (peherstorfer@wisc.edu).

§Department of Aeronautics & Astronautics, MIT, Cambridge, USA kwillcox@mit.edu).

this division, and modifies the subspace during the evaluation process to better meet the current conditions for the reduced state vector prediction.

Online subspace adaptation can be approached from a geometric perspective: The set of all subspaces $\mathcal{U} \subset \mathbb{R}^n$ of a certain fixed dimension p forms the Grassmann manifold [2]. Subspaces are spatial locations on this manifold and are represented in numerical schemes by column-orthogonal matrices in $\mathbb{R}^{n\times p}$. One-parameter subspace modifications correspond to curves on the Grassmannian.

In the special case, where the subspace adaptation is based on a linear least-squares residual function, the *Grassmannian Rank-One Update Subspace Estimation* (GROUSE, [7]) applies: When approximating an unsampled state vector in the subspace $\mathcal{U}$ based on partial information, then the associated least-squares residual is related to a velocity vector of a geodesic curve on the Grassmannian. GROUSE shows that this geodesic curve corresponds to a matrix curve of rank-one modifications on the underlying column-orthogonal matrices that act as subspace representatives.

Main contributions. We show that the GROUSE geodesic of rank-one updates crosses a subspace $\mathcal{U}^*$ that allows for an *exact* representation of the given partial information. Mathematically, this is a nonlinear root-finding problem on the Grassmann manifold. We derive a closed-form expression for the residual with respect to the partial information along the GROUSE geodesic. In particular, this allows us to read off the root, but it may be of potential use in general when analyzing GROUSE with other step size schemes. As an auxiliary, we establish a general formula for the rank-one update of orthogonal projectors. Moreover, we generalize the method to subspace adaptation based on general least-squares systems and to the adaptation of a subspace of the subspace in question.

In the results section, we demonstrate that the proposed method applies in combination with the following well-established dimension reduction techniques: gappy proper orthogonal decomposition (gappy POD, [27, 17]) and discrete empirical interpolation method (DEIM, [21]). More precisely, we consider an application to gappy POD image processing, and we combine the subspace adaptation with the DEIM to construct an adaptive reduced model for the time-dependent nonlinear FitzHugh-Nagumo partial differential equation system, which models the electrical activity in a neuron. In contrast to the standard use case in the GROUSE literature [7, 49], our focus is not on estimating a subspace from scratch based on potentially noisy data but to adapt a given subspace of valid approximations based on incomplete but noise-free observations. In the DEIM setting, it is not the final subspace that is of main interest but rather the enhanced approximation capabilities after each adaptation.

Context and related work. The Grassmann manifold can be represented as a matrix manifold. For comprehensive background information on optimization on matrix manifolds, we refer to [2] and its extensive bibliography. Matrix manifolds appear frequently in image processing and computer vision [35], where they often take the form of subspace identification problems. A related field of application is low-rank matrix factorizations, which arise in data analysis problems of various kinds, among them matrix completion [7], [14]. The GROUSE method was introduced in [7] as a tool for both subspace identification from incomplete and/or noisy data and the matrix completion problem and was further developed and analyzed in [9, 31, 48, 49].

A recent survey of model reduction methods for parametric systems is [13]. Most online adaptive model reduction techniques rely on pre-computed quantities that restrict the way the reduced space can be changed online. One example is parametric model reduction based on the interpolation of reduced models, where reduced operators are interpolated on matrix manifolds [3, 23, 38, 4, 36, 50]. There are also dictionary approaches [30, 34] that construct a reduced space online from a subset of a large number of pre-computed basis vectors, and localized reduced modeling techniques [5, 40, 26, 24] that select online one of several pre-computed reduced models.

In contrast, we are here interested in online adaptive model reduction methods that derive updates to the reduced model with information that is obtained from the full model in the online phase; thus, the adaptation uses information that is unavailable in the offline phase. There are several approaches that generate new data from the full model in the online phase, or derive new reduced basis vectors with an h-refinement [20] based on an adjoint model of the full model, and then rebuild the reduced model [37, 39, 44, 45]; however, this is often computationally expensive. An efficient online adaptation that uses new data online was presented in [46, 22] for localized reduced models. A reference state is subtracted from the snapshots of localized reduced models. It is shown that this corresponds to a rank-one update of the reduced space corresponding to the localized reduced models; however, this is only a limited form of adapting a reduced model because each snapshot receives the same change. In [42, 41], dynamic reduced models are introduced that adapt to changes in the full model without requiring access to the high-fidelity operators; however, the approach is limited to linear problems and to problems where high-resolution sensor information is available that provides approximations of the full state vectors. For nonlinear problems, an adaptive DEIM was presented in [43], which derives low-rank updates to the DEIM basis from sparse data of nonlinear terms. In this paper we draw on the theory of Grassman manifolds and subspace updates to introduce a more flexible method for adaptive model reduction that applies to nonlinear problems and reproduces the inputed sparse data exactly.

Notation and preliminaries. The $(p \times p)$-*identity matrix* is denoted by $I_p \in \mathbb{R}^{p\times p}$. If the dimension is clear, we will simply write I. The $(p \times p)-$*orthogonal group*, i.e., the set of all square orthogonal matrices, is denoted by

$$O_p = \{R \in \mathbb{R}^{p\times p} | R^T R = RR^T = I_p\}.$$

For a matrix $U \in \mathbb{R}^{n\times p}$, the subspace spanned by the columns of U is denoted by $\mathcal{U} := \operatorname{colspan}(U) := \{U\alpha \in \mathbb{R}^n | \quad \alpha \in \mathbb{R}^p\} \subset \mathbb{R}^n$. The set of all p-dimensional subspaces $\mathcal{U} \subset \mathbb{R}^n$ forms the *Grassmann manifold*

$$Gr(n, p) := \{\mathcal{U} \subset \mathbb{R}^n | \quad \mathcal{U} \text{ subspace}, \dim(\mathcal{U}) = p\}.$$

The *Stiefel manifold* is the compact matrix manifold of all column-orthogonal rectangular matrices

$$St(n, p) := \{U \in \mathbb{R}^{n\times p} | \quad U^T U = I_p\}.$$

The Grassmann manifold can be realized as a quotient manifold of the Stiefel manifold

$$Gr(n, p) = St(n, p)/O_p = \{[U] | \quad U \in St(n, p)\}, \tag{1}$$

where $[U] = \{UR | \quad R \in O_p\}$ is the *orbit*, or *equivalence class* of U under actions of the orthogonal group. Hence, by definition, two matrices $U, \tilde{U} \in St(n, p)$ are in the same O_p-orbit if they differ by a $(p \times p)$-orthogonal matrix:

$$[U] = [\tilde{U}] :\Leftrightarrow \exists R \in O_p : U = \tilde{U}R.$$

A matrix $U \in St(n, p)$ is called a *matrix representative* of a subspace $\mathcal{U} \in Gr(n, p)$, if $\mathcal{U} = \operatorname{colspan}(U)$. We will also consider the orbit $[U]$ and the subspace $\mathcal{U} = \operatorname{colspan}(U)$

as the same object. As in [25], we will make use throughout of the quotient representation (1) of the Grassmann manifold with matrices in $St(n,p)$ acting as representatives in numerical computations. From the manifold perspective, each p-dimensional subspace of $\mathbb{R}^n$ is a *single point* on $Gr(n,p)$.

For a rectangular, full column rank matrix $X \in \mathbb{R}^{n\times p}$, the *orthogonal projection* onto the column span of X is

$$\Pi_X : \mathbb{R}^n \to \operatorname{colspan} X, \quad y \mapsto X(X^TX)^{-1}X^Ty. \tag{2}$$

We will consider special orthogonal projectors associated with the Cartesian coordinate directions. Let $e_j \in \mathbb{R}^n$ denote the jth canonical unit vector, $j = 1,\dots,n$. Given a subset of $m \in \mathbb{N}$ indices $J = \{j_1,\dots,j_m\} \subset \{1,\dots,n\}$, the (column-orthogonal) matrix $P = (e_{j_1},\dots,e_{j_m}) \in \{0,1\}^{n\times m}$ is called *the mask matrix corresponding to the index set* J. Left-multiplication of a vector with the transpose of P realizes the projection onto the selected components in the same order as listed in J, i.e., $P^Ty = (y_{j_1},\dots,y_{j_m})^T \in \mathbb{R}^m$ for all $y \in \mathbb{R}^n$. The matrix PP^T is the canonical orthogonal projection onto the coordinate axes $j_1,\dots,j_m$.

Throughout, whenever a mask matrix $P \in \mathbb{R}^{n\times m}$ is applied to a subspace representative $U \in St(n,p)$, we assume that $m > p$ and that the matrix of selected rows $P^TU \in \mathbb{R}^{m\times p}$ has full column rank p.

Organization. Section 2 recaps the GROUSE approach and transfers the idea of the geometric subspace adaptation to the context of model reduction. It also reviews the essentials on the numerical treatment of Grassmann manifolds. Section 3 presents the core methodological contributions of this paper, where we derive a closed-form of the Grassmann rank-one update that solves the underlying least-squares residual equation exactly. Example applications in the context of adaptive model reduction and image processing are presented in Section 4, and Section 5 concludes the paper.

2. Problem statement.

In this section, we first summarize GROUSE following Ref. [7]. We then develop the connection between the theory of GROUSE and the task of adapting a low-dimensional subspace for model reduction. Lastly, we discuss relevant concepts in the numerical treatment of Grassmann manifolds.

2.1. GROUSE.

Let $P = (e_{j_1},\dots,e_{j_m}) \in \{0,1\}^{n\times m}$ be a mask matrix, let $\mathcal{U}_0 \subset \mathbb{R}^n$ be a p-dimensional subspace with matrix representation $\mathcal{U}_0 = [U_0]$, $U_0 \in St(n,p)$ and let $b \in \mathbb{R}^m$ be a given data vector, $p < m < n$. GROUSE considers the masked least-squares problem

$$y(\mathcal{U}_0) := \underset{\tilde{y}\in\mathcal{U}_0}{\arg\min} \|P^T\tilde{y} - b\|_2^2, \tag{3}$$

which features the (subspace dependent) unique solution

$$y(\mathcal{U}_0) = U_0\alpha(\mathcal{U}_0) \in \mathbb{R}^n, \quad \alpha(\mathcal{U}_0) = (U_0^TPP^TU_0)^{-1}U_0^TPb \in \mathbb{R}^p. \tag{4}$$

The corresponding residual vector $r(\mathcal{U}_0) := b - P^Ty(\mathcal{U}_0)$ is, in general, non-zero. For a fixed mask matrix P and a fixed right-hand side b, the residual vector is associated with a differentiable function on $Gr(n,p)$, the residual norm function

$$F_{P,b} : Gr(n,p) \to \mathbb{R}, \quad \mathcal{U} \mapsto \|r(\mathcal{U})\|_2^2 = b^Tb - b^TP^TU(U^TPP^TU)^{-1}U^TPb. \tag{5}$$

see [7, eq. (2), (3)]. (The matrix U in the definition of $F_{P,b}$ can be any representative $U \in St(n,p)$ of the subspace $\mathcal{U}$, see (1). The subscripts P,b will be dropped, when clear from the context.) Given a sequence of incomplete observations in form of data vectors $b_s \in \mathbb{R}^m$, $s = 1, 2, \dots$ with corresponding mask matrices P_s, GROUSE adapts the initial subspace such that the objective

$$\mathcal{U} \mapsto \sum_{s=1}^{\infty} F_{P_s,b_s}(\mathcal{U}) = \sum_{s=1}^{\infty} \|P_s^T y(\mathcal{U}) - b_s\|^2 \tag{6}$$

is minimized, see [7, eq. (5)].[1]

The GROUSE algorithm works sequentially by addressing one data vector b_s at a time. It performs a step along the geodesic line on $Gr(n, p)$ [25], [2, §4] in the direction of steepest descent, which is given by the negative of the gradient of (5) with respect to the subspace $\mathcal{U}_0 = [U_0]$. The gradient is represented by the rank-one matrix $G = -2P\left(b_s - P^T U_0 \alpha_s\right)\alpha_s^T$ with $\alpha_s = (U_0^T P P^T U_0)^{-1} U_0^T P b_s$, see [7, eq. (9)], [25, eq. (2.70)]. The direction of steepest descent is $H = -G$. Because H is rank-one, its thin SVD $H = \Phi\Sigma V^T$ reduces to $H = \frac{Pr}{\|r\|}(\sigma_1)v^T$, where r is the residual vector, $v = \frac{\alpha_s}{\|\alpha_s\|}$ and $\sigma_1 = 2\|r\|\|\alpha\|$ is the single non-zero singular value of H. Evaluating the Grassmann geodesic [25, §2.5.1] along this descent direction leads to

$$t \mapsto U_0(t) = U_0 + \left((\cos(t\sigma_1) - 1)U_0 v + \sin(t\sigma_1)\frac{Pr}{\|r\|}\right) v^T =: U_0 + \hat{x}(t)v^T, \tag{7}$$

see [7, eq. (11), (12)]. At each iteration $s = 1, 2, \dots$, the GROUSE algorithm [7, Alg. 1] chooses a step size $t = \eta_s$ and replaces the previous subspace representative U_{s-1} by $U_s = U_{s-1}(\eta_s)$ according to (7). Local and global convergence results are given in [8, 48, 49].

2.2. Subspace adaptation and model reduction. We consider here projection-based model reduction methods. These methods make use of a subspace $\mathcal{U}_0 \subset \mathbb{R}^n$ of comparatively low dimension $\dim(\mathcal{U}_0) = p \ll n$ that is assumed to contain the essential information about a set $\mathcal{X} \subset \mathbb{R}^n$ of state vectors over a range of operating conditions. More precisely, the fundamental assumption underlying the dimension reduction is that the n-dimensional state vectors $y \in \mathcal{X}$ may be approximated up to sufficient accuracy with only p degrees of freedom via

$$y \approx \tilde{y}(\alpha) = U_0\alpha, \quad \alpha \in \mathbb{R}^p, \tag{8}$$

where $U_0 \in St(n, p)$ is a matrix representative of $\mathcal{U}_0$. The standard case in model reduction is that the set of state vectors $\mathcal{X}$ is the solution manifold of a parametric partial differential equation (PDE).

In the following, we consider the special case that only incomplete information on a state vector $y \in \mathcal{X}$ is available. This case is encountered in the model reduction techniques gappy POD [27] and DEIM [21]. The incomplete data imposes equality constraints on the $m < n$ components $y_{j_1}, \dots, y_{j_m}$ of a state vector $y \in \mathcal{X}$ via the equation

$$P^T y = \begin{pmatrix} y_{j_1} \\ \vdots \\ y_{j_m} \end{pmatrix} =: b, \quad P = (e_{j_1}, \dots, e_{j_m}) \in \{0,1\}^{n\times m}. \tag{9}$$

[1] For complete data vectors $b_s \in \mathbb{R}^n$, (6) is the same as [49, eq. (2)].

Under the requirement that y be contained in $\mathcal{U}_0$, the underdetermined equation (9) translates into the overdetermined masked least-squares problem (3) with corresponding solution (4). This establishes a direct link to the GROUSE approach.

The objective of our work is to find a subspace $\mathcal{U}^* \in Gr(n,p)$ close to $\mathcal{U}_0$ such that the best subspace-restricted least-squares solution $y(\mathcal{U}^*)$ features an *exact zero residual*, $\|r(\mathcal{U}^*)\|_2 = 0$. In solving this equation for the unknown $\mathcal{U}^*$, we adapt the original reduced subspace $\mathcal{U}_0$ according to the least-squares problem arising from the new (partial) information about y. The requirement of $\mathcal{U}^*$ being close to $\mathcal{U}_0$ is important in the context of model reduction because we want the approximation (8) to remain valid for $\mathcal{U}^*$.

We formalize the objective. Define the *feasibility set*

$$\mathcal{Z} := \{\mathcal{U} \in Gr(n,p) | \quad \min_{\tilde{y}\in\mathcal{U}} \|P^T\tilde{y} - b\|_2 = 0\}. \tag{10}$$

The set $\mathcal{Z}$ is non-empty.[2] From GROUSE, it is known that the geodesic curve $t \mapsto \mathcal{U}(t)$ that starts in $\mathcal{U}(0) = \mathcal{U}_0$ with velocity given by the direction of steepest descent of the residual norm function (5) is a matrix curve of rank-one updates on the initial subspace $\mathcal{U}_0$, see (7). We will show that this curve crosses the feasibility set $\mathcal{Z}$ and determine the first intersection point. By writing the residual vector as $r(\mathcal{U}_0) = b - \Pi_{P^TU_0}b$, where $\Pi_{P^TU_0}$ is the orthogonal projection (2) onto colspan(P^TU_0), this objective becomes a *nonlinear equation* on the Grassmann manifold:

$$\text{solve } b - \Pi_{P^TU(t^*)}b = 0 \text{ for } t^* \in \mathbb{R}. \tag{11}$$

The condition $b - \Pi_{P^TU(t^*)}b = 0$ is equivalent to $[U(t^*)] \in \mathcal{Z}$.

A contribution of this paper is an explicit formula for the time-dependent residual $r(\mathcal{U}(t)) = b - \Pi_{P^TU(t)}b$ derived in Section 3, from which the solution to (11) can be read off in closed form. In contrast to GROUSE, whose overall aim is the iterative global minimization of (6), we focus on the single adaptation steps and the nonlinear residual equation on $Gr(n,p)$. We arrive in this way at the same formula for t^* that was obtained in [49, Alg. 1, §3.1, App. C] as the optimal greedy step size in an iterative subspace updating scheme based on complete right-hand side vectors.

In summary, our approach is a method for determining a subspace $\mathcal{U}^*$ contained in the set $\mathcal{Z}$ from (10) that can be reached via a geodesic path along the descent direction starting in $\mathcal{U}_0$. Figure 1 below and Section S1 from the supplement illustrate this principle. In Subsection 3.3, we show that this is not restricted to the special case of masked least-squares problems $\|P^T\tilde{y} - b\|_2$ but can be generalized to arbitrary underdetermined systems $\|A\tilde{y} - b\|_2$, $A \in \mathbb{R}^{m\times n}$.

2.3. Numerical aspects of the Grassmann manifold. Our approach to solve (11) is presented in Section 3 and builds on geometric concepts on the Grassmann manifold $Gr(n,p)$. This subsection reviews a few essential aspects of the numerical treatment of Grassmann manifolds. We refer to [1, 2, 25] for details.

Tangent spaces and normal coordinates. The *tangent space* $T_\mathcal{U}Gr(n,p)$ at a point $\mathcal{U} \in Gr(n,p)$ can be thought of as the space of velocity vectors of differentiable curves on $Gr(n,p)$ passing through $\mathcal{U}$. For any matrix representative $U \in St(n,p)$ of $\mathcal{U} \in Gr(n,p)$ the tangent space of $Gr(n,p)$ at $\mathcal{U}$ is represented by

$$T_\mathcal{U}Gr(n,p) = \left\{\Delta \in \mathbb{R}^{n\times p} | \quad U^T\Delta = 0\right\} \subset \mathbb{R}^{n\times p},$$

[2] Any subspace $\mathcal{U}$ that contains a vector $y = Pb + v$, where $v \in \mathbb{R}^n$ is in the $(n-m)$-dimensional kernel of P^T is in $\mathcal{Z}$.

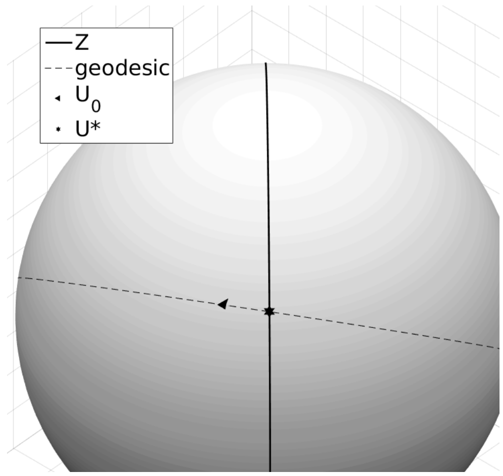


FIG. 1. *Graphical illustration of the geometric subspace adaptation: The sphere visualizes the Grassmann manifold $Gr(n,p)$. The solid line marks the set $\mathcal{Z}$ of all subspaces in $Gr(n,p)$ that contain zero-residual solutions to the least-squares problem* (3)*. The black triangle shows the initial subspace $\mathcal{U}_0$. The dashed line is the geodesic starting in $\mathcal{U}_0$ with velocity given by minus the gradient of the least-squares residual function. Our goal is to compute the subspace $\mathcal{U}^*$, where the geodesic meets the set $\mathcal{Z}$.*

its canonical metric being $\langle \Delta, \tilde{\Delta} \rangle_{Gr} = tr(\Delta^T \tilde{\Delta})$, [25, §2.5]. Endowing each tangent space with this metric turns $Gr(n,p)$ into a *Riemannian manifold*. A geodesic $t \mapsto \mathcal{U}(t)$ on $Gr(n,p)$ is a locally length-minimizing curve. A geodesic is uniquely determined by its starting point $\mathcal{U}(0)$ and its starting velocity $\dot{\mathcal{U}}(0) = \Delta \in T_{\mathcal{U}_0}Gr(n,p)$, [2, p. 102].

The corresponding Riemannian *exponential mapping* is

$$Exp_{\mathcal{U}_0} : T_{\mathcal{U}_0}Gr(n,p) \to Gr(n,p), \quad \Delta \mapsto Exp_{\mathcal{U}_0}(\Delta) := \mathcal{U}(1).$$

The Riemannian exponential maps a tangent vector $\Delta \in T_{\mathcal{U}_0}Gr(n,p)$ to the endpoint $\mathcal{U}(1)$ of a geodesic path $\mathcal{U} : [0,1] \to Gr(n,p)$ starting at $\mathcal{U}(0) = \mathcal{U}_0 \in Gr(n,p)$ with velocity $\Delta \in T_{\mathcal{U}_0}Gr(n,p)$.

An efficient algorithm for evaluating the Grassmann exponential is derived in [25, §2.5.1]. The explicit form of the associated geodesic is

$$\mathcal{U}(t) = Exp_{\mathcal{U}_0}(t\Delta) = [U_0 V \cos(t\Sigma)V^T + \Phi \sin(t\Sigma)V^T], \quad \Delta \overset{\text{SVD}}{=} \Phi\Sigma V^T. \tag{12}$$

The exponential mapping gives a local parametrization from the (flat, Euclidean) tangent space to the manifold. This is also referred to as to representing the manifold in *normal coordinates* [32, §III.8], [33, Lem. 5.10].

Distance between subspaces. Given two subspaces $[U], [\tilde{U}] \in Gr(n,p)$, the ith *canonical* or *principal angle* between $[U]$ and $[\tilde{U}]$ is $\theta_i := \arccos(\sigma_i) \in [0, \frac{\pi}{2}]$, where σ_i is the ith-largest singular value of $U^T\tilde{U} \in \mathbb{R}^{p\times p}$ [29, §12.4.3].

The Riemannian distance between $[U], [\tilde{U}] \in Gr(n,p)$ is

$$\text{dist}([U], [\tilde{U}]) := \|\Theta\|_2, \quad \Theta = (\theta_1, \dots, \theta_p) \in \mathbb{R}^p. \tag{13}$$

Normal coordinates are radially isometric with respect to the Riemannian distance on $Gr(n,p)$ and the canonical metric on $T_{\mathcal{U}_0}Gr(n,p)$ in the following sense: the length of a tangent vector Δ as measured by the metric in $T_{\mathcal{U}_0}Gr(n,p)$ is the same as the Riemannian distance $\mathrm{dist}(\mathcal{U}_0, Exp_{\mathcal{U}_0}(\Delta))$ on $Gr(n,p)$, provided that Δ is in a neighborhood of $0 \in T_{\mathcal{U}_0}Gr(n,p)$, where the exponential is invertible, [33, Lem. 5.10 & Cor. 6.11].

The Grassmann manifold is a compact homogeneous space [32]. In particular, by [47, Thm 8(b)], any two points on $Gr(n,p)$ can be connected by a geodesic of length $\leq \frac{\sqrt{p}}{2}\pi$. This is related to the so-called *injectivity radius* of the Grassmann manifold [47], which is the maximal radius ρ such that the exponential map at *any point* $[U] \in Gr(n,p)$ is a diffeomorphism onto the open ball $B(0,\rho) \subset T_{[U]}Gr(n,p)$ around the origin in the corresponding tangent space. The injectivity radius of the Grassmann manifold is $\rho = \frac{\pi}{2}$, [47]. This concept is relevant to the step of conducting the line search within Grassmann optimization schemes. We make the following observation: Using the explicit formulas for the exponential mapping and its (local) inverse, called the logarithmic mapping $Log_{[U]}$, see [11, §3], one can show that $Log_{[U]} \circ Exp_{[U]}(\Delta) = \Delta$ for all tangent vectors Δ of *spectral* norm $\|\Delta\|_2 = \sigma_1(\Delta) < \pi/2$, where $\sigma_1(\Delta)$ is the largest singular value of Δ. As a consequence, we have

Observation 1. For all $[U] \in Gr(n,p)$, let

$$B_{[U],spec}(0,\pi/2) := \left\{ \Delta \in T_{[U]}Gr(n,p) | \quad \sigma_1(\Delta) < \frac{\pi}{2} \right\}.$$

Then the exponential mapping $Exp_{[U]}$ is a radial isometry on $B_{[U],spec}(0,\pi/2)$.

This observation is important for numerical computations because

$$B_{[U],spec}(0,\pi/2) \supset \left\{ \Delta \in T_{[U]}Gr(n,p) | \quad \sqrt{\langle \Delta, \Delta \rangle_{Gr}} = \|(\sigma_1, \ldots, \sigma_p)^T\|_2 < \frac{\pi}{2} \right\},$$

i.e., the spectral $\pi/2$-ball in the tangent space encloses the canonical $\pi/2$-ball in the tangent space. The above observation leads to the next proposition which has implications on the uniqueness of solutions to (11).

PROPOSITION 1. *Let $[U] \in Gr(n,p)$, $\Delta \in T_{[U]}Gr(n,p)$ and $\tilde{U} = Exp_{[U]}(\Delta)$. If $\|\Delta\|_2 < \frac{\pi}{2}$, then dist $\Big([U], [\tilde{U}]\Big) = \|\Delta\|_{Gr}$. In particular, the length of the geodesic path starting in $[U]$ and ending in $[\tilde{U}]$ is less than $\frac{\sqrt{p}}{2}\pi$.*

Proof. Let $\Delta \overset{\text{SVD}}{=} \Phi\Sigma V^T$ with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$ and $\sigma_1 = \|\Delta\|_2 < \frac{\pi}{2}$. The exponential projection of Δ onto $Gr(n,p)$ is $[\tilde{U}] = Exp_{[U]}(\Delta) = [UV\cos(\Sigma)V^T + \Phi\sin(\Sigma)V^T]$.

The SVD of $U^T\tilde{U}$ is $V\cos(\Sigma)V^T$, so that $0 \leq \theta_k := \arccos(\cos(\sigma_k)) = \sigma_k < \frac{\pi}{2}$. Hence, $(\sigma_1, \ldots, \sigma_p)^T = (\theta_1, \ldots, \theta_p)^T := \Theta \in \mathbb{R}^p$ is precisely the vector of canonical angles between $[U]$ and $[\tilde{U}]$ (when listing the canonical angles in descending order), see (13). As a consequence,

$$\mathrm{dist}\Big([U], [\tilde{U}]\Big) = \|\Theta\|_2 = \sqrt{tr(\Sigma^2)} = \sqrt{tr(\Delta^T\Delta)} = \|\Delta\|_{Gr}.$$

Since $\sigma_1 < \frac{\pi}{2}$, we have $\|\Delta\|_{Gr} = \left(\sum_{i=1}^p \sigma_p^2\right)^{1/2} < \frac{\sqrt{p}}{2}\pi$. ∎

A subtlety of Proposition 1 is that the length condition on Δ is with respect to the spectral norm rather than the canonical norm.

3. Solving the Grassmann residual equation. We now return to our goal formulated in Subsection 2.2: the solution of eq. (11) . In Subsection 3.1, we derive a general update formula for orthogonal projectors under rank-one modifications. Subsection 3.2 derives an explicit time-dependent expression for the Grassmann residual along the GROUSE geodesic. In particular, this allows us to read off the closed-form solution to (11). A generalization to least-squares systems featuring arbitrary matrices rather than mask matrices as operators is given in Subsection 3.3. Subsequently, Subsection 3.4 introduces an extension for performing the Grassmann subspace adaptation over selected directions of the subspace only.

3.1. A closed-form rank-one update for orthogonal projectors. In this subsection, we derive a formula for orthogonal projectors under rank-one updates that turns out to be an essential building block in solving (11). As this result is also of independent interest, we state it in a more general setting.

Let $X \in \mathbb{R}^{m\times p}$. Recall from (2) that the orthogonal projection onto colspan X is $\Pi_X = X(X^TX)^{-1}X^T$. Let $X \overset{\text{SVD}}{=} Q\Sigma R^T$ be the thin SVD of X with $Q \in St(m,p)$, $\Sigma \in \mathbb{R}^{p\times p}$ diagonal, $R \in O_p$ orthogonal. Then Π_X is expressed alternatively as $\Pi_X = QQ^T$.

Let $x \in \mathbb{R}^m$, $v \in \mathbb{R}^p$ and consider the rank-one update

$$X_{new} = X + xv^T \in \mathbb{R}^{m\times p}.$$

We are interested in an expression $\Pi_{X_{new}} = Q_{new}Q_{new}^T$, where $Q_{new} \in St(m,p)$. One standard way to approach this is via rank-one SVD updates, [18, 16]. However, this requires an auxiliary SVD of a $(p \times p)$-matrix. Here, we can avoid this, since we are not interested in the fully updated $X_{new} \overset{\text{SVD}}{=} Q_{new}\Sigma_{new}R_{new}^T$ or even in Q_{new} alone but only in $Q_{new}Q_{new}^T$.

LEMMA 2. *As in the above setting, let $X \overset{SVD}{=} Q\Sigma R^T$, $X_{new} = X + xv^T$ and define*

$$\tilde{q} = x - QQ^Tx, \quad q = \frac{\tilde{q}}{\|\tilde{q}\|_2} \in \mathbb{R}^m, \tag{14a}$$

$$g = \begin{pmatrix} g_p \\ g_{p+1} \end{pmatrix} = \begin{pmatrix} -\Sigma^{-1}R^Tv \\ \frac{1}{\|\tilde{q}\|_2}(1 + x^TQ\Sigma^{-1}R^Tv) \end{pmatrix} \in \mathbb{R}^{p+1}. \tag{14b}$$

Then the orthogonal projection onto colspan(X_{new}) *is*

$$\Pi_{X_{new}} = (Q,q)\begin{pmatrix} Q^T \\ q^T \end{pmatrix} - \frac{1}{\|g\|_2^2}(Q,q)gg^T\begin{pmatrix} Q^T \\ q^T \end{pmatrix}. \tag{15}$$

Proof. We start with a decomposition inspired by [16, eq. (3)]. Note that $(Q,q) \in St(m,p+1)$ by construction. It holds that

$$X + xv^T = (Q,q)\begin{pmatrix} \Sigma R^T + Q^Txv^T \\ \|\tilde{q}\|v^T \end{pmatrix} =: (Q,q)M,$$

where $M \in \mathbb{R}^{(p+1)\times p}$. Let $M \overset{\text{SVD}}{=} \tilde{Q}\tilde{\Sigma}\tilde{R}^T$ be the thin SVD of M, i.e., $\tilde{Q} \in St(p+1,p), \tilde{\Sigma}, \tilde{R}^T \in \mathbb{R}^{p\times p}$. Formally, the updated SVD is

$$X + xv^T = \left((Q,q)\tilde{Q}\right)\tilde{\Sigma}\tilde{R}^T =: Q_{new}\Sigma_{new}R_{new}^T.$$

Let $g \in \mathbb{R}^{p+1}$ be such that $(\tilde{Q}, \frac{g}{\|g\|}) \in O_{p+1}$ is an orthogonal completion of $\tilde{Q}$. Because of $I_{p+1} = (\tilde{Q}, \frac{g}{\|g\|})(\tilde{Q}, \frac{g}{\|g\|})^T$, we have

$$\tilde{Q}\tilde{Q}^T = I_{p+1} - \frac{1}{\|g\|^2} g g^T$$

and, as a consequence,

$$Q_{new}Q_{new}^T = (Q, q)\tilde{Q}\tilde{Q}^T \begin{pmatrix} Q^T \\ q^T \end{pmatrix} = (Q, q)\left(I_{p+1} - \frac{1}{\|g\|^2} g g^T\right)\begin{pmatrix} Q^T \\ q^T \end{pmatrix}. \tag{16}$$

Hence, it is sufficient to determine g, which is characterized up to a scalar factor by $\tilde{Q}^T g = 0$. Since $\operatorname{colspan}(M) = \operatorname{colspan}(\tilde{Q})$, this condition is equivalent to $M^T g = 0$. Let $g_p \in \mathbb{R}^p$ denote the first p components of g and let $g_{p+1} \in \mathbb{R}$ be the last entry such that $g^T = (g_p^T, g_{p+1})$. When writing the equation $g^T M = 0$ as

$$(g_p^T, g_{p+1})\begin{pmatrix} \Sigma & Q^T x \\ 0 & \|\tilde{q}\|_2 \end{pmatrix}\begin{pmatrix} R^T \\ v^T \end{pmatrix} = 0,$$

it is straightforward to show that $g = \begin{pmatrix} -\Sigma^{-1}R^T v \\ \frac{1}{\|\tilde{q}\|_2}(1 + x^T Q\Sigma^{-1}R^T v) \end{pmatrix} \in \mathbb{R}^{p+1}$ and any scalar multiple of this vector is a valid solution. Using this vector in (16) proves the lemma. ∎

3.2. An explicit expression for the Grassmann residual function along the GROUSE geodesic. We now state our main theorem on the solution of the nonlinear equation (11) .

THEOREM 3. *Let $\mathcal{U}_0 = [U_0] \in Gr(n, p)$ be represented by $U_0 \in St(n, p)$. Let $P = (e_{j_1}, \ldots, e_{j_m}) \in \{0, 1\}^{(n\times m)}$ be a mask matrix. Moreover, let $b \in \mathbb{R}^m$ and suppose that $U_0^T Pb \neq 0$.*

Let $\alpha = (U_0^T PP^T U_0)^{-1} U_0^T Pb$ be the optimal coefficient vector corresponding to the masked least-squares problem

$$\min_{\tilde{\alpha}\in\mathbb{R}^p} \|P^T U_0 \tilde{\alpha} - b\|^2$$

and let $r = b - P^T U_0 \alpha$ the associated residual vector. Set $v = \frac{\alpha}{\|\alpha\|_2}$ and $s_1 = 2\|r\|_2\|\alpha\|_2$. Moreover, write $P^T U_0 \overset{SVD}{=} Q\Sigma R^T \in \mathbb{R}^{m\times p}$ and $g_p = -\Sigma^{-1}R^T v$. The t-dependent residual vector along the geodesic descent direction is

$$r([U(t)]) = b - \Pi_{P^T U(t)} b = \frac{\|r\|_2 - \|\alpha\|_2 \tan(ts_1)}{1 + \tan^2(ts_1)\|g_p\|^2}\left(\frac{r}{\|r\|_2} - \frac{\tan(ts_1)}{\|\alpha\|_2} Q\Sigma^{-2}Q^T b\right).$$

Proof. Reconsider (7) and let

$$x(t) = P^T\hat{x}(t) = (\cos(ts_1) - 1)P^T U_0 v + \sin(ts_1)\frac{r}{\|r\|_2},$$

$$v = \frac{\alpha}{\|\alpha\|_2}, \quad \alpha = (U_0^T PP^T U_0)^{-1}U_0^T Pb,$$

so that

$$P^T U(t) = P^T U_0 + x(t)v^T.$$

Since $P^TU(t)$ is a rank-one update of P^TU_0, Lemma 2 applies. Introducing $P^TU_0 \overset{\text{SVD}}{=} Q\Sigma R^T \in \mathbb{R}^{m\times p}$, we obtain $r = b - QQ^Tb$ and $\alpha = R\Sigma^{-1}Q^Tb$. The t-dependent orthogonal projection onto colspan$(P^TU(t))$ is

$$\Pi_{P^TU(t)} = (Q, q(t))\begin{pmatrix} Q^T \\ q^T(t)\end{pmatrix} - \frac{1}{\|g(t)\|_2^2}(Q, q(t))g(t)g^T(t)\begin{pmatrix} Q^T \\ q^T(t)\end{pmatrix}, \tag{18}$$

where

$$\tilde{q}(t) = x(t) - QQ^Tx(t), \quad q(t) = \frac{\tilde{q}(t)}{\|\tilde{q}(t)\|_2} \in \mathbb{R}^m,$$

$$g(t) = \begin{pmatrix} g_p \\ g_{p+1}(t)\end{pmatrix} = \begin{pmatrix} -\Sigma^{-1}R^Tv \\ \frac{1}{\|\tilde{q}(t)\|_2}(1 + x^T(t)Q\Sigma^{-1}R^Tv)\end{pmatrix} \in \mathbb{R}^{p+1}.$$

We have $Q^Tr = 0$ and thus $Q^Tx(t) = \frac{\cos(ts_1)-1}{\|\alpha\|_2}Q^Tb$. This leads to $\tilde{q}(t) = \frac{\sin(t)}{\|r\|_2}r$ and $\|\tilde{q}(t)\|_2 = |\sin(t)|$ as well as $q(t) = \text{sign}(\sin(t))\frac{r}{\|r\|_2} = \pm q$, were we standardize $q = \frac{r}{\|r\|_2}$. Moreover,

$$x^T(t)Q\Sigma^{-1}R^Tv = \frac{1}{\|\alpha\|_2^2}(\cos(ts_1) - 1)\underbrace{b^TQ\Sigma^{-2}Q^Tb}_{\|\alpha\|_2^2} = (\cos(ts_1) - 1),$$

so that $g(t)$ is

$$g(t) = \begin{pmatrix} -\frac{1}{\|\alpha\|_2}\Sigma^{-2}Q^Tb \\ \frac{\cos(ts_1)}{|\sin(ts_1)|}\end{pmatrix} \in \mathbb{R}^{p+1}.$$

It holds $\frac{\cos(ts_1)}{|\sin(ts_1)|}q(t) = \frac{\cos(ts_1)}{\sin(ts_1)}q$. Hence, according to (18), we may consistently work with $+q$ and $\frac{\cos(ts_1)}{\sin(ts_1)} = \cot(ts_1)$. In order to evaluate the updated projection (18), we compute

$$(Q, q)\begin{pmatrix} g_p \\ g_{p+1}(t)\end{pmatrix} = -\frac{1}{\|\alpha\|_2}Q\Sigma^{-2}Q^Tb + \cot(ts_1)q,$$

$$g_p^TQ^Tb = -\frac{1}{\|\alpha\|_2}b^TQ\Sigma^{-2}Q^Tb = -\|\alpha\|_2 \text{ and}$$

$$q^Tb = \frac{1}{\|r\|_2}r^Tb = \frac{1}{\|r\|_2}(\underbrace{b^Tb - b^TQQ^Tb}_{\|r\|_2^2}) = \|r\|_2.$$

Substituting these identities in (18), we arrive at

$$\begin{aligned} r([U(t)]) = b - \Pi_{P^TU(t)}b &= b - QQ^Tb - qq^Tb \\ &+ \frac{1}{\|g(t)\|_2}\left(Qg_p + \cot(ts_1)q\right)\left(g_p^TQ^Tb + \cot(ts_1)q^Tb\right) \\ &= \frac{\cot(ts_1)\|r\|_2 - \|\alpha\|_2}{\|g(t)\|_2^2}\left(\cot(ts_1)\frac{r}{\|r\|_2} - \frac{1}{\|\alpha\|_2}Q\Sigma^{-2}Q^Tb\right), \end{aligned} \tag{20}$$

as was claimed. ∎

Note that the only special property of P that is exploited in the proof is that $P^TPr = r$. Hence, the result holds when P is replaced with an arbitrary column-orthogonal matrix.

There is a number of conclusions that can be drawn from Theorem 3:

COROLLARY 4. 1. *The t-dependent residual norm along the steepest descent direction is*

$$\|r([U(t)])\|_2 = \|b - \Pi_{P^T U(t)} b\|_2 = \frac{|\|r\|_2 - \|\alpha\|_2 \tan(ts_1)|}{\sqrt{1 + \|g_p\|_2^2 \tan^2(ts_1)}}. \tag{21}$$

2. *The residual norm function is continuous and $\frac{\pi}{s_1}$-periodic along the steepest descent direction with*

$$\|r([U(0)])\|_2 = \|r\|_2 \text{ and } \|r([U\left(\frac{\pi}{2s_1}\right)])\|_2 = \frac{\|\alpha\|_2}{\|g_p\|_2} = \frac{\|\alpha\|_2^2}{\|Q\Sigma^{-2}Q^T b\|_2}.$$

3. *The first root along the geodesic descent direction is at*

$$t^* = \frac{1}{s_1} \arctan\left(\frac{\|r\|_2}{\|\alpha\|_2}\right) \in \left(0, \frac{\pi}{2s_1}\right). \tag{22}$$

The associated matrix $U^* := U_0 + \left((\cos(t^* s_1) - 1)U_0 v + \sin(t^* s_1)\frac{Pr}{\|r\|}\right) v^T$ is such that the subspace $\mathcal{U}^* := [U^*]$ is contained in the set $\mathcal{Z}$ from (10), i.e.,

$$F(\mathcal{U}^*) = \min_{\tilde{\alpha} \in \mathbb{R}^p} \|P^T U^* \tilde{\alpha} - b\|^2 = 0. \tag{23}$$

Stated differently, it holds that b is contained in colspan $(P^T U^*)$, *that is, $b = \Pi_{P^T U^*} b$.*

4. *The coefficient vector associated with $\mathcal{U}^* = [U^*] = (23)$ is $\alpha^* = \sqrt{1 + \frac{\|r\|_2^2}{\|\alpha\|_2^2}}\alpha$. The associated $y^* \in \mathbb{R}^n$ is $y^* = U^*\alpha^* = U_0\alpha + Pr = U_0\alpha + P(b - P^T U_0 \alpha)$. Hence, y^* can be readily obtained without computing any of t^*, α^*, U^*.*

5. *The first maximum along the geodesic descent direction is at*

$$t_{max} = \frac{1}{s_1}\left(\pi - \arctan\left(\frac{\|\alpha\|_2}{\|r\|_2 \|g_p\|_2^2}\right)\right) \in \left(\frac{\pi}{2s_1}, \frac{\pi}{s_1}\right)$$

with corresponding value $\|r([U(t_{max})])\|_2 = \sqrt{\|r\|_2^2 + \frac{\|\alpha\|_2^2}{\|g_p\|_2^2}}$.

Proof. By taking into account that r is orthogonal to colspan(Q), Pythagoras' Theorem gives $\| \left(\cot(ts_1)\frac{r}{\|r\|_2} - \frac{1}{\|\alpha\|_2} Q\Sigma^{-2}Q^T b\right) \|_2 = \sqrt{\cot^2(ts_1) + \|g_p\|_2^2} = \|g(t)\|_2$. The formula (21) is now an immediate consequence of (20). From (21), the statements 2., 3., and 5. of the corollary are straightforward.
On statement 4.: From 3., we know that there exists $\alpha^* \in \mathbb{R}^p$ such that $P^T U^* \alpha^* - b = 0$. After plugging in the explicit expression for U^*, we obtain the equation

$$P^T U_0 \left(\alpha^* - \frac{\alpha^T \alpha^*}{\|\alpha\|_2^2}\alpha\right) + \left(\frac{\alpha^T \alpha^*}{\|\alpha\|_2 \sqrt{\|\alpha\|_2^2 + \|r\|_2^2}} - 1\right) b = 0.$$

If the unmodified least-squares problem (3) features a nonzero residual, then b is not contained in colspan $P^T U_0$. Hence, both quantities in the round brackets must be zero, which leads to $\alpha^* = \frac{\alpha^T\alpha^*}{\|\alpha\|_2^2}\alpha = \frac{\sqrt{\|\alpha\|_2^2 + \|r\|_2^2}}{\|\alpha\|_2}\alpha$. The calculation of y^* is straightforward. ∎

Appendix A features a short cut to statements 3. and 4. of Corollary 4. An example of a plot of the residual norm function (21) from a practical application is displayed in Figure 5.

Remark 5. The GROUSE convergence analysis in [9] is based on local considerations and a step length of $\tilde{t} = \frac{1}{s_1} \arcsin\left(\frac{\|r\|_2}{\|\alpha\|_2}\right)$, which matches the t^* in (22) up to terms of third order, when the residual and therefore the ratio $\|r\|_2/\|\alpha\|_2$ is small. In the fully sampled case, that is, when complete right-hand side data is available, Ref. [49] shows that the same t^* of (22) is also the greedy-optimal step with respect to the determinant-similarity and the Frobenius norm discrepancy of two subspaces in an iterative subspace updating scheme, see [49, §3.1 & App. C]. In contrast, we arrived at t^* from the independent approach of solving the nonlinear equation (11) and with a different proof that relies on Lemma 2. Combining these facts shows that the subspace discrepancy is maximal if and only if the subspace update is such that the residual vanishes exactly.

The proof of Proposition 1 shows that the distance between the subspaces $[U_0]$ and $[U^*]$ is $t^* s_1 = \arctan\left(\frac{\|r\|_2}{\|\alpha\|_2}\right) < \frac{\pi}{2}$. Hence, when performing the t^*-optimal rank-one update on $[U_0]$ according to Corollary 4, we stay within the injectivity radius. As a consequence from general differential geometry, the geodesic $t \mapsto [U(t)]$ is length-minimizing, that is, there is no shorter curve on $Gr(n,p)$ that connects $[U_0]$ and $[U^*]$.[3]

We emphasize that the update formula of Lemma 2 for orthogonal projectors under rank-one modifications was used as an intermediate theoretical fact in proving Theorem 3 but that it is not required to actually compute the rank-one update and the associated quantities Q, q, g in order to obtain the optimal t^* and the subspace $[U^*] = [U(t^*)]$. MATLAB code that considers this fact is in the supplement in Section S4.

We draw a corollary that corresponds to the special case where the mask matrix P is the identity I_n, i.e., the case where complete data is available. Recall that the best least-squares approximation to a given vector b that is contained in a subspace $\mathcal{U}_0$ is the orthogonal projection $U_0 U_0^T b$ of b onto $\mathcal{U}_0$, with an associated residual of $r = b - U_0 U_0^T b$. The SVD of $P^T U_0$ is now trivially $P^T U_0 = Q\Sigma R^T = U_0 I_p I_p^T$ so that the expressions involving Q, Σ, R simplify.

COROLLARY 6. *Let $\mathcal{U}_0 = [U_0] \in Gr(n,p)$ be represented by $U_0 \in St(n,p)$. Let $b \in \mathbb{R}^n$ and suppose that $\alpha := U_0^T b \neq 0$. Set $v = \frac{\alpha}{\|\alpha\|_2}$ and $s_1 = 2\|r\|_2\|\alpha\|_2$. Then the t-dependent residual norm is*

$$\|r([U(t)])\|_2 = \|b - \Pi_{U(t)} b\|_2 = \frac{|\|r\|_2 - \|\alpha\|_2 \tan(ts_1)|}{\sqrt{1 + \tan^2(ts_1)}}.$$

Define

$$t^* = \frac{1}{s_1} \arctan\left(\frac{\|r\|_2}{\|\alpha\|_2}\right).$$

Then $U^* := U(t^*) := U_0 + \left((\cos(t^* s_1) - 1)U_0 v + \sin(t^* s_1)\frac{r}{\|r\|}\right) v^T$ is such that b is contained in the subspace $\mathcal{U}^* := [U^*]$, i.e., $b = \Pi_{U^*} b$.

Remark 7. Corollary 6 has a connection with rank-one SVD updates as considered in [18, 15, 16]. One application in [16, Table 1] is to *revise* an existing SVD $U_0 \Sigma_0 V_0^T = (X, c)$ such that the column c is replaced with a column b in the modified SVD $U' \Sigma' V'^T = (X, b)$. In terms of the associated orthogonal projectors, we have $U' U'^T b = b$. With Corollary 6, we obtain a subspace $[U^*]$ that also contains b. Yet,

[3] This does not necessarily mean that there is no other point $[\tilde{U}^*] \in \mathcal{Z}$ that is closer to $[U_0]$.

this is not achieved by explicitly exchanging a column c of the original data matrix for the new column b. Rather, via the update $U_0+\left((\cos(t^*s_1)-1)U_0v+\sin(t^*s_1)\frac{r}{\|r\|}\right)v^T$ $=: U_0+x^*v^T$, the missing residual part is distributed over all columns of the original representative U_0. In order to emulate this with the 'revise'-approach of [16, Table 1], one first has to rotate the subspace representative with $\Phi=(v^\perp,v)\in O_p$, so that $(U_0+x^*v^T)\Phi=U_0\Phi+(0,\ldots,0,x^*)$, i.e., the rank-one update acts on a single direction of the new representative $U_0\Phi$. Allowing for rotations of the representative U_0 in the update scheme enables more general updates than when working with a fixed representative U_0. Hence, we expect that $\mathrm{dist}([U_0],[U^*])\leq \mathrm{dist}([U_0],[U'])$. This is confirmed in the example featured in Subsection 4.2.

Another relation between GROUSE and the incremental SVD of [15] was exposed in [8]. The approach considered in [8] corresponds to first attaching new column data to a given subspace representative. Then, the SVD update is performed on the augmented matrix representative and consequently retruncated to its original dimensions. It is shown that this procedure can be emulated via GROUSE when a specific step size is chosen for the rank-one increment. However, the modified U' obtained in this way does not feature the property $U'U'^Tb=b$, i.e., it does not correspond to a subspace that reproduces b exactly. More details can be found in Section S2.

3.3. The general case. When the operator in the underlying least-squares problem (3) is not a mask matrix but an arbitrary real matrix, then the Grassmann gradient associated with the residual function is still rank-one so that GROUSE continues to apply. Convergence results for GROUSE with arbitrary sampling matrices are given in [48].

Mind that Corollary 4 remains valid with the same proof, when the mask matrix P is replaced with an arbitrary column-orthogonal matrix. For general subspace-restricted least-squares problems

$$\min_{\tilde\alpha\in\mathbb{R}^p}\|AU_0\tilde\alpha-b\|^2,$$

where the operator $A\in\mathbb{R}^{m\times n}$, $m\leq n$ is arbitrary but such that AU_0 has full column rank, we can proceed as follows. Let $QR=A^T$ be the thin qr-decomposition of A^T with $Q\in St(n,m)$, $R\in\mathbb{R}^{p\times p}$. Then

$$\|AU_0\tilde\alpha-b\|^2=\|R^T\left(Q^TU_0\tilde\alpha-(R^T)^{-1}b\right)\|^2$$

Since Q is column-orthogonal, we may apply Theorem 3, Corollary 4 to the least-squares problem

$$\min_{\tilde\alpha}\|Q^TU_0\tilde\alpha-(R^T)^{-1}b\|^2$$

to produce a modified U^* such that $\hat\alpha:=\arg\min_{\tilde\alpha\in\mathbb{R}^p}\|Q^TU^*\tilde\alpha-(R^T)^{-1}b\|^2$ fulfills $0=\|Q^TU^*\hat\alpha-(R^T)^{-1}b\|^2$. As a consequence, $\|AU^*\hat\alpha-b\|^2=0$. In summary:

THEOREM 8. *Let $p<m\leq n$. Consider the general subspace restricted least-squares problem*

$$\min_{\tilde\alpha\in\mathbb{R}^p}\|AU_0\tilde\alpha-b\|^2,\quad A\in\mathbb{R}^{m\times n},\quad b\in\mathbb{R}^m,\quad [U_0]\in Gr(n,p),\quad \mathit{rank}(AU_0)=p.$$

Let $QR=A^T$ and suppose that R is regular. Then there exists a subspace $[U^*]\in$

$Gr(n,p)$ such that

$$\min_{\tilde{\alpha}\in\mathbb{R}^p} \|AU^*\tilde{\alpha} - b\|^2 = 0 \text{ and } dist([U_0],[U^*]) = \arctan\left(\frac{\|r\|_2}{\|\alpha\|_2}\right) = \tau^*,$$

where $\alpha = \arg\min_{\tilde{\alpha}\in\mathbb{R}^p} \|Q^T U_0\tilde{\alpha} - (R^T)^{-1}b\|^2$, $r = (R^T)^{-1}b - Q^T U_0\alpha$. *The subspace U^* is given by*

$$U^* = U_0 + \left((\cos(\tau^*) - 1)U_0\frac{\alpha}{\|\alpha\|_2} + \sin(\tau^*)\frac{Qr}{\|r\|_2}\right)\frac{\alpha^T}{\|\alpha\|_2}.$$

3.4. Adapting a subspace of a subspace. There are many applications where it might be desirable to keep some directions of a given subspace fixed while adapting the remaining ones. In the context of adaptive model reduction, such situations are likely to occur if the columns spanning the subspace in question stem from a principal component analysis or proper orthogonal decomposition (POD), and are thus ordered by information content. In these cases, the user might want to keep the most dominant subspace directions fixed, while adapting the portion of the subspace spanned by the less important basis vectors. This subsection describes the modifications to the methodology for doing so, a sample application is presented in Subsection 4.2.

Let $f : Gr(n,p) \to \mathbb{R}, [U] \mapsto f([U])$ be a differentiable function. Let us divide the column set of a subspace representative $U \in St(n,p)$ into a constant portion $U_c \in St(n,p-l)$ and a portion $U_l \in St(n,l)$ that is subject to change, so that $U = (U_c, U_l) \in St(n,p-l)\times St(n,l)$. By fixing U_c, we obtain a function $f_l : Gr(n,l) \to \mathbb{R}, f_l([U_l]) = f([U_c,U_l])$ with gradient $G_l := \nabla f_l([U_l]) \in \mathbb{R}^{n\times l}$. The gradient induces the search direction $H_l = -G_l$. The geodesic associated with the search direction $H_l \overset{\text{SVD}}{=} \Phi_l S_l V_l^T \in \mathbb{R}^{n\times l}$ is represented by

$$U_l(t) = Exp_{[U_l]}(tH_l) = U_l V_l \cos(tS_l)V_l^T + \Phi_l \sin(tS_l)V_l^T. \tag{24}$$

Note that S_l and V_l are $(l\times l)$-matrices. For each t, the matrix $U_l(t) \in St(n,l)$ is a feasible *orthogonal* subspace representative. Yet, we have to consider the possibility that the compound matrix $(U_c, U_l(t))$ ceases to be a valid subspace representative in $St(n,p)$.[4] It is even conceivable that $[U_l(t)]$ moves towards the subspace $[U_c]$ spanned by the fixed basis vectors so that the compound matrix $(U_c, U_l(t))$ not only loses the orthogonal-columns property but even becomes rank deficient. One way to avoid this, is to re-orthogonalize $U_l(t)$ against U_c, say, by conducting an extra Gram-Schmidt procedure. However, Proposition 9 below implies that the orthogonality between the columns of the matrices $U_l(t)$ and the constant columns of the matrix block U_c is preserved along the geodesic path in *direction of the least-squares gradient*, so that in this case, the corresponding compound matrix $(U_c, U_l(t))$ is also an orthogonal matrix representative in $St(n,p)$ and a Gram-Schmidt re-orthogonalization is unnecessary.

PROPOSITION 9. *Let $f : Gr(n(,p) \to \mathbb{R}$ be differentiable. Suppose that*

$$T_{[U]}Gr(n,p) \ni \quad \nabla_{[U]} f = \left(\nabla_{[U_c]} f_c, \nabla_{[U_l]} f_l\right) \quad \in \left(T_{[U_c]}Gr(n,p-l)\right)\times\left(T_{[U_l]}Gr(n,l)\right), \tag{25}$$

where it is understood that $\nabla_{[U_c]} f_c$ and $\nabla_{[U_l]} f_l$ denote the gradients of the restrictions $f_c : [U_c] \mapsto f([U_c,U_l])$ and $f_l : [U_l] \mapsto f([U_c,U_l])$, respectively.

Let $[U_0] = [(U_c, U_{l,0})] \in Gr(n,p)$. Let $t \mapsto [U_l(t)] \subset Gr(n,l)$ be the geodesic path along the descent direction $-\nabla_{[U_{l,0}]} f_l$. Then $U_c^T U_l(t) = 0$ for all t.

[4] Appendix B shows that this actually may happen even along search directions of rank one.

Therefore, the corresponding curve of concatenated matrices $(U_c, U_l(t)) \subset \mathbb{R}^{n\times p}$ is a curve of orthogonal matrices in $St(n,p)$. Hence, for each t, $[(U_c, U_l(t))] \in Gr(n,p)$, in consistency with the quotient space view point (1).

Proof. Let $U_0 = (U_c, U_{l,0}) \in St(n,p)$, where $U_c \in St(n, p-l)$ and $U_{l,0} \in St(n,l)$. The gradient with respect to f is a tangent vector in $T_{[U_0]}Gr(n,p)$, hence $U_0^T \nabla_{[U_0]} f = 0$. By (25),

$$0 = U_0^T \nabla_{[U_0]} f = \begin{pmatrix} U_c^T \\ U_{l,0}^T \end{pmatrix} \left(\nabla_{[U_c]} f_c, \nabla_{[U_{l,0}]} f_l\right). \tag{26}$$

In particular, $U_c^T \nabla_{[U_{l,0}]} f_l = 0$. Writing $\nabla_{[U_{l,0}]} f_l \overset{\text{SVD}}{=} -\Phi_l S_l V_l^T \in \mathbb{R}^{n\times l}$, we have $U_c^T \Phi_l = 0$, since the columns of Φ_l span the same space as the columns of $\nabla_{[U_l]} f_l$. Hence, the geodesic at t, $U_l(t) = U_{l,0} V_l \cos(tS_l) V_l^T + \Phi_l \sin(tS_l) V_l^T$ is also orthogonal to U_c, i.e., $U_c^T U_l(t) = 0$. ◻

As can be seen from the proof, the proposition is not specific to the GROUSE context nor does it depend on the rank of the gradient. It holds in general, whenever the gradient splitting of (25) holds. This, however, is not always the case, see Appendix B. The objective function F of (5) features this property: When allowing only the last l directions of (U_c, U_l) to vary, we obtain a differentiable $F_l : Gr(n,l) \to \mathbb{R}$ with

$$F_l([U_l]) = b^T b - b^T P^T (U_c, U_l)\left(\begin{pmatrix} U_c^T \\ U_l^T \end{pmatrix} P P^T (U_c, U_l)\right)^{-1} \begin{pmatrix} U_c^T \\ U_l^T \end{pmatrix} Pb.$$

The associated gradient, now a rank-one $(n \times l)$-matrix, reads

$$G_l := \nabla_{[U_l]} F_l = -2P\left(b - P^T U\alpha\right)\alpha^T \begin{pmatrix} 0_{(p-l)\times l} \\ I_l \end{pmatrix}, \quad \alpha = (U^T P P^T U)^{-1} U^T P b,$$

where $U = (U_c, U_l)$. The next corollary transfers the result of Corollary 4 to the setting of adapting only the last l columns of a given subspace representative.

COROLLARY 10. *Let $\mathcal{U}_0 = [U_0] \in Gr(n,p)$ be represented by $U_0 \in St(n,p)$. Let $P = (e_{j_1}, \ldots, e_{j_m}) \in \{0,1\}^{(n\times m)}$ be a mask matrix and let $b \in \mathbb{R}^m$.*

Let $\alpha = (U_0^T P P^T U_0)^{-1} U_0^T P b$ be the optimal coefficient vector corresponding to the masked least-squares problem

$$\min_{\tilde{\alpha}\in\mathbb{R}^p} \|P^T U_0 \tilde{\alpha} - b\|^2$$

and let $r = b - P^T U_0 \alpha$ be the associated residual vector. Let $l \in \mathbb{N}$, $l \leq p$ and write column-wise $U_0 = (U_c, U_{l,0})$, $U_c = \left(u_0^1, \ldots, u_0^{p-l}\right)$, $U_{l,0} = \left(u_0^{p-l+1}, \ldots, u_0^p\right)$. Moreover, let $\alpha_l = \left(0_{(p-l)\times l}, I_l\right)\alpha$ and $v_l = \frac{\alpha_l}{\|\alpha_l\|_2} \in \mathbb{R}^l$.

Set $s_1 = 2\|r\|_2\|\alpha_l\|_2$ and define

$$t^* = \frac{1}{s_1}\arctan\left(\frac{\|r\|_2}{\|\alpha_l\|_2}\right)$$

and $U_l(t^*) = U_{l,0} + \left((\cos(t^* s_1) - 1) U_l v_l + \sin(t^* s_1)\frac{Pr}{\|r\|}\right) v_l^T$.

Then $U^* := U(t^*) := (U_c, U_l(t^*))$ is such that the subspace $\mathcal{U}^* := [U^*]$ is contained in the set $\mathcal{Z}$ from (10), i.e.,

$$F(\mathcal{U}^*) = \min_{\tilde{\alpha}\in\mathbb{R}^p} \|P^T U^* \tilde{\alpha} - b\|^2 = 0,$$

which means that t^* solves (11) .

Proof. According to Proposition 9, the concatenated matrix $(U_c, U_l(t))$ is a valid subspace representative in $St(n,p)$ for all t. Applying the mask operator P to $(U_c, U_l(t))$ leads to the matrix curve

$$P^T U(t) = P^T(U_c, U_l(t)) = P^T(U_c, U_l V_l \cos(tS_l)V_l^T + \Phi_l \sin(tS_l)V_l^T).$$

Because Φ_l, S_l, V_l stem from an SVD of the rank-one gradient $-G_l$, we have that $S_l = \mathrm{diag}(s_1, 0, \dots, 0)$, $s_1 = 2\|r\|_2\|\alpha_l\|_2$. It follows that

$$\begin{aligned} P^T U(t) &= P^T(U_c, U_{l,0}) + \left(0_{n\times(p-l)}, (\cos(ts_1) - 1)P^T U_l v_l + \sin(ts_1)\tfrac{r}{\|r\|_2}\right) \\ &= P^T(U_c, U_{l,0}) + x(t)\left(0_{1\times(p-l)}, v_l^T\right), \end{aligned}$$

where $v_l = \frac{\alpha_l}{\|\alpha_l\|_2}$ is the first column of V_l. This is again a rank-one update on $P^T U(t)$ and the rest of the proof is analogous to the proof of Theorem 3. ∎

Remark 11. When we are adapting only the last column u_0^p of the initial matrix $U_0 = (u_0^1, \dots, u_0^p) \in St(n,p)$, then the resulting U^* is given by $(u_0^1, \dots, u_0^{p-1}, u_0^p(t^*))$, where the last column evaluates to $u_0^p(t^*) = \frac{1}{\sqrt{\|r\|_2^2 + |\alpha_p|^2}}(u_0^p\alpha_p + Pr)$. This is precisely the same result that is obtained by replacing the last column of U_0 with the vector $U_0\alpha + Pr$ and re-orthogonalization the new last column against the columns of $U_0^{p-1} := (u_0^1, \dots, u_0^{p-1})$ via a single Gram-Schmidt step $(I - U_0^{p-1}(U_0^{p-1})^T)(U_0\alpha + Pr) = (u_0^p\alpha_p + Pr)$. In this case, the t^*-GROUSE update applied to the last column of the subspace representative U_0 and the ($[U]$-part of the) 'revise' SVD update of [16, Table 1, p.23] coincide, cf. Remark 7. For more details, see Section S2.

4. Application to adaptive model reduction. This section applies the geometric rank-one subspace update in the specific contexts of online adaptive model reduction and image reconstruction. For each application, we describe how the subspace adaptation is employed and we demonstrate the method with numerical examples.

4.1. Adaptation for POD-DEIM reduced models. We present an online adaptive DEIM that is based on our geometric rank-one subspace update. In contrast to the standard use case in the GROUSE literature [7, 49], the focus here is not on estimating a subspace from scratch based on a global objective function (6) but to adapt a subspace that is already a good approximant for the underlying simulation process during the online phase.

We first formulate our online adaptive DEIM for nonlinear dynamical systems and then present numerical results for the FitzHugh-Nagumo system. To ease exposition and to focus on benchmarking our online adaptive DEIM reduced models, we consider dynamical systems without parameters and inputs. Thus, the aim of the following reduced models is to reproduce well the solution of the full-order dynamical system, instead of predicting solutions for new parameters and inputs. We note, however, that the following POD-DEIM and our online adaptive POD-DEIM reduced models are applicable to parametrized models and models with inputs, see [43, 13].

4.1.1. POD-DEIM-Galerkin reduced models. Consider a nonlinear dynamical system in the time interval $[0,T] \subset \mathbb{R}$, with end time $T > 0$. Let $t_0, t_1, \dots, t_K \in [0,T] \subset \mathbb{R}$ be $K+1 \in \mathbb{N}$ time steps with $t_0 = 0$ and $t_K = T$. Discretizing with, e.g., the forward Euler method leads to the system of equations

$$Ey_i = Ay_{i-1} + f(y_{i-1}), \qquad i = 1, \dots, K, \tag{28}$$

corresponding to the time steps $t_1, \dots, t_K$, respectively. Let $n \in \mathbb{N}$ denote the dimension of the discrete state space. We have the system matrices $A \in \mathbb{R}^{n \times n}$ and $E \in \mathbb{R}^{n \times n}$. The nonlinear function $f : \mathbb{R}^n \to \mathbb{R}^n$ corresponds to the nonlinear terms of the dynamical system. The state vector at time step t_i is denoted as $y_i \in \mathbb{R}^n$. The initial condition is $y_0 \in \mathbb{R}^n$. We consider here the case where the nonlinear function f is evaluated componentwise at the state vector y_i, see, e.g., [21]. We further assume the well-posedness of (28).

To derive a reduced model of the full model (28), we select a set of $n_s \in \mathbb{N}$ snapshots $\{y_{j_1}, \dots, y_{j_{n_s}}\} \subset \{y_1, \dots, y_K\}$ at the time steps $t_{j_1}, \dots, t_{j_{n_s}}$ with indices $j_1, \dots, j_{n_s} \in \{1, \dots, K\}$. POD constructs orthonormal basis vectors $v_1, \dots, v_{n_r} \in \mathbb{R}^n$ of the n_r-dimensional POD space that is the solution to the minimization problem

$$\min_{v_1,\dots,v_{n_r} \in \mathbb{R}^n} \sum_{i=1}^{n_s} \left\| y_{j_i} - \sum_{l=1}^{n_r} (v_l^T y_{j_i}) v_l \right\|_2^2 .$$

The POD basis $V = (v_1, \dots, v_{n_r})$ is formed of the left-singular vectors of the snapshot matrix $Y = (y_{j_1}, \dots, y_{j_{n_s}}) \in \mathbb{R}^{n \times n_s}$ corresponding to the n_r largest singular values. The POD-Galerkin reduced model of (28) is

$$\tilde{E}\tilde{y}_i = \tilde{A}\tilde{y}_{i-1} + V^T f(V\tilde{y}_{i-1}) , \tag{29}$$

where $\tilde{y}_i \in \mathbb{R}^{n_r}$ is the reduced state vector at time step t_i for $i = 1, \dots, K$, and $\tilde{E} = V^T E V, \tilde{A} = V^T A V$ are the reduced operators.

Solving (29) requires evaluating the nonlinear function $f(V\tilde{y}_{i-1})$ at the n-dimensional vector $V\tilde{y}_{i-1} \in \mathbb{R}^n$, which can be computationally expensive. DEIM derives an approximation of $f(V\tilde{y}_{i-1})$ to avoid evaluating f at all n components of $V\tilde{y}_{i-1}$. To this end, DEIM constructs $p \in \mathbb{N}$ DEIM basis vectors $u^1, \dots, u^p \in \mathbb{R}^n$ using POD on the nonlinear snapshots $f(y_{j_1}), \dots, f(y_{j_{n_s}}) \in \mathbb{R}^n$. The DEIM basis vectors are the columns of the DEIM basis matrix $U = (u_1, \dots, u_p) \in \mathbb{R}^{n \times p}$. Additionally, DEIM selects $p \in \mathbb{N}$ DEIM interpolation points $q_1, \dots, q_p \in \{1, \dots, n\}$ using a greedy strategy, see [21]. The DEIM mask matrix is $P = (e_{q_1}, \dots, e_{q_p}) \in \{0, 1\}^{n \times p}$. The DEIM interpolant is the pair (U, P). The DEIM approximation of the nonlinear function f evaluated at the vector $V\tilde{y}_i$ is given as

$$f(V\tilde{y}_i) \approx U(P^T U)^{-1} P^T f(V\tilde{y}_i) . \tag{30}$$

The POD-DEIM-Galerkin reduced model of (28) at a time step $t_i, i = 1, \dots, K$ is

$$\tilde{E}\tilde{y}_i = \tilde{A}\tilde{y}_{i-1} + V^T U(P^T U)^{-1} P^T f(V\tilde{y}_{i-1}) . \tag{31}$$

The reduced model (31) is often orders of magnitude faster to solve than the full model (28) and the reduced state vectors $\tilde{y}_1, \dots, \tilde{y}_K \in \mathbb{R}^{n_r}$ lead to accurate approximations $V\tilde{y}_1, \dots, V\tilde{y}_K \in \mathbb{R}^n$ of the full state vectors $y_1, \dots, y_K \in \mathbb{R}^n$, respectively.

4.1.2. Online adaptive model reduction. We adapt the DEIM interpolant of the nonlinear function f in the online phase, i.e., we adapt the DEIM basis U and the DEIM mask matrix P during the time stepping. We proceed as follows. Let U_0 denote the DEIM basis matrix, which is derived using POD as discussed in Section 4.1.1. Let further $q_1^0, \dots, q_p^0 \in \{1, \dots, n\}$ be the DEIM interpolation points and $P_0 = (e_{q_1^0}, \dots, e_{q_p^0})$ the mask matrix that are derived with the DEIM procedure in the offline phase, see Section 4.1.1. Consider now the online phase at time step t_1.

To compute the reduced state vector $\tilde{y}_1$, we first adapt the DEIM basis matrix U_0 and the mask matrix P_0 to U_1 and P_1, respectively, and then use the adapted DEIM interpolant (U_1, P_1) in the reduced model (31) to compute the reduced state vector $\tilde{y}_1$. The DEIM basis matrix U_0 is adapted to U_1 using the GROUSE rank-one update, as we will discuss in detail in Section 4.1.3. This process is continued iteratively, i.e., at time step t_i, we adapt U_{i-1} and P_{i-1} to obtain U_i and P_i, respectively, and then use the adapted interpolant (U_i, P_i) for computing the reduced state vector $\tilde{y}_i$ at time step t_i. Note that the POD basis matrix V and the reduced linear operators $\tilde{E}$ and $\tilde{A}$ are kept unchanged online (although in principle they too could be adapted).

4.1.3. Subspace adaptation in online adaptive model reduction. We use the GROUSE rank-one update with the residual-annihilating step size (22) to adapt the DEIM basis matrix. Consider time step t_i for $i = 1, \dots, K$. To adapt the DEIM basis matrix U_{i-1} to U_i at time step t_i, we follow [43] and *oversample* the DEIM approximation. Let $\{q^i_{p+1}, \dots, q^i_{p+s}\} \subset \{1, \dots, n\} \setminus \{q^{i-1}_1, \dots, q^{i-1}_p\}$ be a set of $s \in \mathbb{N}$ additional indices that are drawn uniformly from the set $\{1, \dots, n\} \setminus \{q^{i-1}_1, \dots, q^{i-1}_p\}$, where $q^{i-1}_1, \dots, q^{i-1}_p$ are the DEIM interpolation points of the previous time step t_{i-1}. The extended mask matrix $S_i \in \{0,1\}^{n \times m}$, $m = p + s$, is assembled from the points in the set $\{q^{i-1}_1, \dots, q^{i-1}_p, q^i_{p+1}, \dots, q^i_{p+s}\}$ as $S_i = (e_{q^{i-1}_1}, \dots, e_{q^{i-1}_p}, e_{q^i_{p+1}}, \dots, e_{q^i_{p+s}})$. The matrix S_i corresponds to $m = p + s > p$ point indices, and therefore the interpolation problem (30) of the classical DEIM approximation with the interpolant (U_{i-1}, P_{i-1}) becomes an overdetermined least-squares problem using the extended mask matrix S_i

$$\alpha = \underset{\tilde{\alpha} \in \mathbb{R}^p}{\arg\min} \|S_i^T U_{i-1} \tilde{\alpha} - S_i^T f(V\tilde{y}_{i-1})\|_2^2 \tag{32}$$

with

$$f(V\tilde{y}_{i-1}) \approx U_{i-1}\alpha \,.$$

The solution α of (32) is

$$\alpha = (U_{i-1}^T S_i S_i^T U_{i-1})^{-1} U_{i-1}^T S_i S_i^T f(V\tilde{y}_{i-1}) \,.$$

The regression problem (32) fits into the framework of the GROUSE subspace adaptation approach of Subsection 2.2, so that we can find the adapted DEIM basis matrix U_i with the low-rank update derived in Corollary 4. In addition to updating the DEIM basis matrix, the DEIM interpolation points $q^{i-1}_1, \dots, q^{i-1}_p$ are updated to $q^i_1, \dots, q^i_p$. For this task we use the algorithm introduced in [43, Section 4]. The entire DEIM online adaptivity procedure is summarized in Algorithm 1.

4.1.4. Example of DEIM subspace adaptation. We apply the online subspace adaptation to the DEIM interpolant of a reduced model of the FitzHugh-Nagumo system. The FitzHugh-Nagumo system is used in the original DEIM paper [21] as a benchmark example. The number of time steps is $K = 10^6$ and the dimension of the discretized state space is $n = 2048$. The state vectors $y_0, y_{1000}, y_{2000}, \dots, y_K \in \mathbb{R}^n$ at every 1000th time step are used as snapshots to construct $n_r = 10$ POD basis vectors and the corresponding POD basis matrix $V \in \mathbb{R}^{n \times n_r}$. The nonlinear function is evaluated at the snapshot time instances to obtain the nonlinear snapshots $f(y(t_0)), f(y(t_{1000})), f(y(t_{2000})), \dots, f(y(t_K))$.

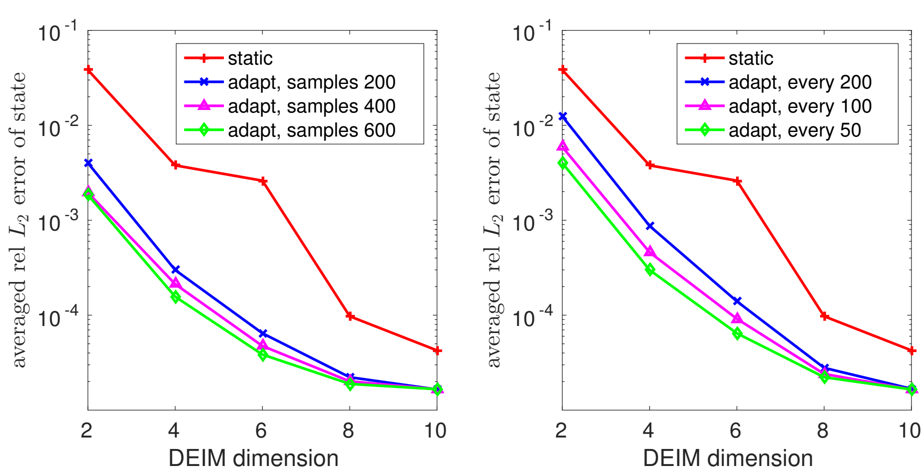
We compare the error of a static reduced model without online subspace adaptation to the error of an adaptive reduced model as in Alg. 1. We report the average of the relative L_2 error of the approximation $V\tilde{y}_i$ to the reference y_i at the time steps

Algorithm 1 Time stepping a reduced model with online adaptive DEIM

Input: System matrices E, A, nonlinear function f, initial condition y_0, POD basis matrix V, DEIM basis matrix U_0, DEIM interpolation points matrix P_0, number of sampling points s, adaptation interval l

Set $\tilde{y}_0 = V^T y_0$
for $i = 1, \dots, K$ **do**
  if $\mod(i, l) == 0$ **then**
    {Adapt DEIM interpolant every l-th time step}
    Set $q_1^{i-1}, \dots, q_p^{i-1}$ to the interpolation points of P_{i-1}
    Draw $q_{p+1}^i, \dots, q_{p+s}^i$ uniformly from $\{1, \dots, n\} \setminus \{q_1^{i-1}, \dots, q_p^{i-1}\}$
    Construct mask matrix S_i from points $q_1^{i-1}, \dots, q_p^{i-1}, q_{p+1}^i, \dots, q_{p+s}^i$
    Evaluate nonlinear function at sampling points $b = S_i^T f(V\tilde{y}_{i-1})$
    {Employ Corollary 4 to adapt U_{i-1}}
    Set $\alpha = (U_{i-1}^T S_i S_i^T U_{i-1})^{-1} U_{i-1}^T S_i b$, and $r = b - S_i^T U_{i-1}\alpha$
    Set $v = \alpha / \|\alpha\|_2, s_1 = 2\|r\|_2\|\alpha\|_2$, and $t^* = s_1^{-1} \arctan(\|r\|_2/\|\alpha\|_2)$
    Adapt basis matrix
      $U_i = U_{i-1} + ((\cos(t^* s_1) - 1)\, U_{i-1} v + \sin(t^* s_1)(S_i r)/\|r\|_2)\, v^T$
    Adapt interpolation points matrix P_{i-1} to P_i with [43, Algorithm 2]
  else
    Set $U_i = U_{i-1}$ and $P_i = P_{i-1}$ {No adaptation}
  end if
  $\tilde{f}_i = V^T U_i (P_i^T U_i)^{-1} P_i^T f(V\tilde{y}_{i-1})$ {Approximate nonlinear function}
  Solve reduced model $\tilde{E}\tilde{y}_i = \tilde{A}\tilde{y}_{i-1} + \tilde{f}_i$ for $\tilde{y}_i$
end for

Output: Reduced states $\tilde{y}_0, \dots, \tilde{y}_K$



(a) adapt every 50th time step with varying sample size

(b) adapt with $s = 200$ samples with varying adaptation interval

FIG. 2. *The average relative L_2 error of a static reduced model is compared to the error of a reduced model with an online adaptive DEIM interpolant. The online adaptation based on the low-rank updates achieves an up to an order of magnitude improvement in the L_2 error compared to the static DEIM interpolant.*

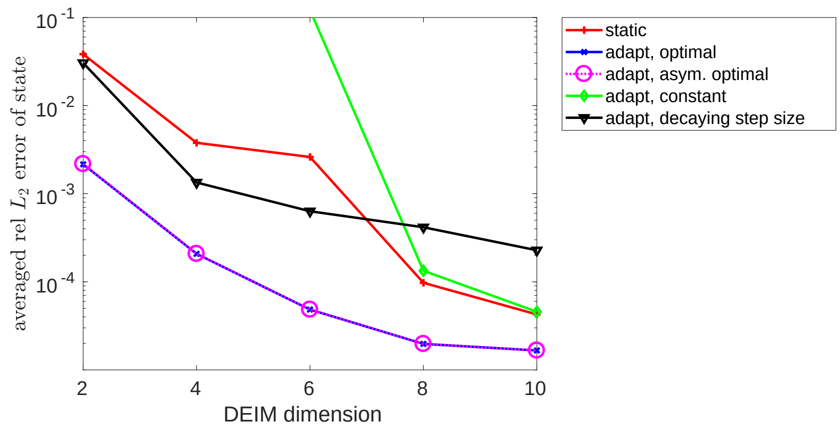


FIG. 3. *The plot reports the error of the online adaptive POD-DEIM reduced model for different step sizes. The label "adapt, optimal" refers to the residual annihilator derived in Corollary 4, "adapt, asym. optimal" refers to the step size $\tilde{t}$ mentioned in Remark 5, "adapt, constant" to the constant step size* 0.05*, and "adapt, decaying step size" to the step size* $0.05/i$*, where* i *is the counter variable in Algorithm 1. Note that the curves of "adapt, optimal" and "adapt, asym. optimal" are on top of each other.*

$t_{500}, t_{1500}, \dots, t_{K-500}$. Thus, the error is measured at time steps other than where the snapshots were taken.

Figure 2(a) compares the L_2 error of the states of the reduced model (31) with a static DEIM interpolant to the error of the reduced model with an adaptive DEIM interpolation. The dimension of the DEIM subspace is varied over the range $p \in \{2, 4, 6, 8, 10\}$. The DEIM subspace and the DEIM interpolation points are adapted every 50th time step, which means that we set $l = 50$ in Alg. 1. At each adaptation step, the geometric rank-one update of Corollary 4 is performed to adapt the DEIM basis matrix based on $s \in \{200, 400, 600\}$ sampling points. Note that the computational costs of the rank-one update are bounded by $\mathcal{O}(np)$. The error of the static and the online adaptive reduced model decreases with the DEIM dimension, which shows that the POD space, which is static and derived from snapshots taken over the whole time interval, approximates well the full-order state vectors, see Subsection 4.1.1. The online adaptive DEIM interpolant can further reduce the error by about an order of magnitude. Figure 2(b) reports results for the online adaptive reduced model, where the DEIM interpolant is adapted every 50th, 100th, and 200th time step with a fixed number of $s = 200$ samples. This means that Algorithm 1 is run with $l = 50, 100, 200$, respectively. The results confirm that increasing the number of adaptivity steps increases the accuracy of the results.

Figure 3 shows results for the online adaptive DEIM interpolant where different step sizes are used. We compare four different step size selections in Figure 3. The curve with the label "adapt, optimal" refers to the residual annihilator t^*, which is derived in Corollary 4 and implemented in Algorithm 1. The curve with label "adapt, asym. optimal" corresponds to the step size $\tilde{t} = \frac{1}{s_1} \arcsin\left(\frac{\|r\|_2}{\|\alpha\|_2}\right)$ that is discussed in the GROUSE convergence analysis of [9], see also Remark 5. We additionally compare to the constant step size 0.05 in "adapt, constant" and a decaying step size $0.05/i$ in "adapt, decaying step size", as in, e.g., the GROUSE numerical experiments in [7], where i is the counter variable in the for-loop in Algorithm 1. The number of samples is set to $s = 400$ and the DEIM subspace and the DEIM interpolation points are adapted every 50th time step. The optimal and the asymptotically optimal step size lead to similar results (the curves are on top of each other), which was to be expected, since the functions arctan and arcsin match up to terms of third order. The less sophisticated choices "adapt, constant" and "adapt, decaying step size" lead to poor results which are even worse than those produced by the static subspace for DEIM basis dimensions of 8 and 10. This shows that for the application at hand, it is crucial to select a residual-related step size based on the ratio $\frac{\|r\|_2}{\|\alpha\|_2}$, e.g., the minimizer t^* from Corollary 4.

4.2. Subspace adaptation for gappy POD image reconstruction. In this section, the geometric subspace update is applied in combination with the method of gappy POD [27, 17] on an image processing problem, where we use the method to implant a new feature into a given subspace.

FIG. 4. *Face database used for gappy POD example.*

We briefly summarize gappy POD. Given a set of snapshots $\{y_k |\ k = 1, \dots, n_s\} \subset \mathbb{R}^n$, let $\mathcal{U} = \text{colspan}(U)$ be the associated POD subspace represented by $U \in St(n, p)$ with $p \leq n_s$. Let further $y^g \in \mathbb{R}^n$ be an *incomplete snapshot* associated with an index set $J = \{j_1, \dots, j_m\} \subset \{1, \dots, n\}$ of cardinality $m \in \mathbb{N}$; y^g is incomplete in the sense that only components with indices in J are considered as accurate information. Gappy POD computes a vector contained in $\mathcal{U}$ that best fits the incomplete snapshot y^g in a least-squares sense. Employing the mask matrix $P = (e_{j_1}, \dots, e_{j_m}) \in \{0, 1\}^{n \times m}$, the gappy POD approximation $y^{gpod} \in \mathbb{R}^n$ is determined by the masked least-squares minimization problem

$$y^{gpod} = U\alpha_{gpod}, \quad \alpha_{gpod} = \arg\min_{\alpha \in \mathbb{R}^p} \|P^T U\alpha - P^T y^g\|^2. \tag{33}$$

(Notice the similarities to the DEIM approach from Section 4.1.4. Ref. [28] exposes further details on the relation between gappy POD and the Empirical Interpolation Method (EIM, [10]), which predates DEIM.) In our concrete example of image processing, the snapshot set is taken from the so-called *Yale Database* [12], see also [19, §5.2].[5] Representing each image as a snapshot vector $y_k \in \mathbb{R}^n$, $n = 4096$, yields a snapshot matrix of dimension $Y \in \mathbb{R}^{4096 \times 10}$. The snapshots are displayed in Fig. 4. The single image with glasses has been deliberately omitted from the snapshot set, so that no picture in the snapshot ensemble features the property 'glasses-on'. The 'glasses'-detail from this picture, displayed in the lower left corner of Fig. 6, acts as a vector of gappy data $y^g \in \mathbb{R}^{4096}$ with $m = 1336$ non-zero entries and corresponding mask matrix P. The gappy POD objective is to find the linear combination of snapshots that comes closest to represent the 'glasses'–feature in a least-squares sense. The resulting image is displayed in the second column of Fig. 6 with the top picture showing the gappy POD solution and the bottom picture showing the reference image

[5]More precisely, we have used row 11 of the set of 165 Yale images in (64×64)-MATLAB format provided by Deng Cai at http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html.

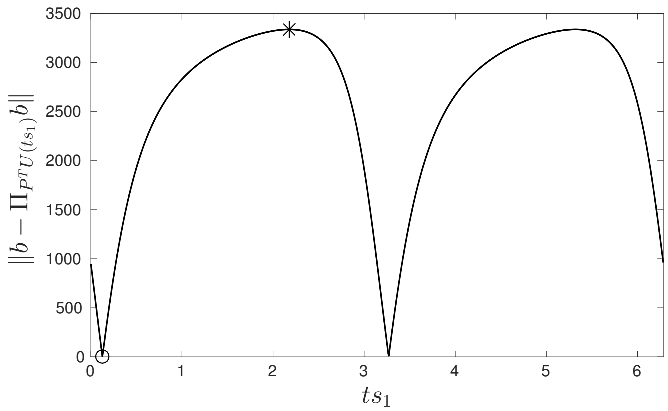


Fig. 5. *Plot of two periods of residual norm function (21) corresponding with the gappy POD subspace adaptation. The circle locates the first root and the star indicates the global maximum.*

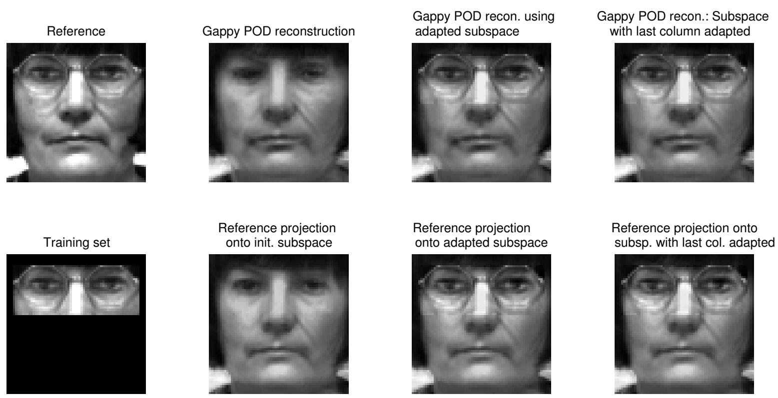


Fig. 6. *Gappy POD approximation of a picture excerpt. To be read column-wise: Reference picture and training excerpt. Gappy POD reconstruction based on the excerpt and projection of complete reference onto the POD subspace. Gappy POD reconstruction using an adapted POD subspace and projection of complete reference thereon. Gappy POD reconstruction after adapting only the last column of POD subspace and projection of complete reference thereon.*

projected onto the subspace spanned by the POD modes. The gappy POD reconstruction is a poor approximation of the reference picture because the POD space does not contain any information required to represent glasses.

Now, we use the GROUSE rank-one update combined with Corollary 4 to annihilate the gappy POD residual, which corresponds to solving the nonlinear equation (11) on $Gr(n,p) = Gr(4096, 10)$. The input data are the mask matrix $P \in \mathbb{R}^{n\times m}$ associated with the picture excerpt, the corresponding right-hand side $b = P^T y^g \in \mathbb{R}^m$, and the subspace representative $U_0 \in St(n,p)$ stemming from a POD of the input snapshots. A plot of the residual norm function along the rank-one update is displayed in Figure 5.

The update leads to a subspace representative U^* that allows for a perfect re-

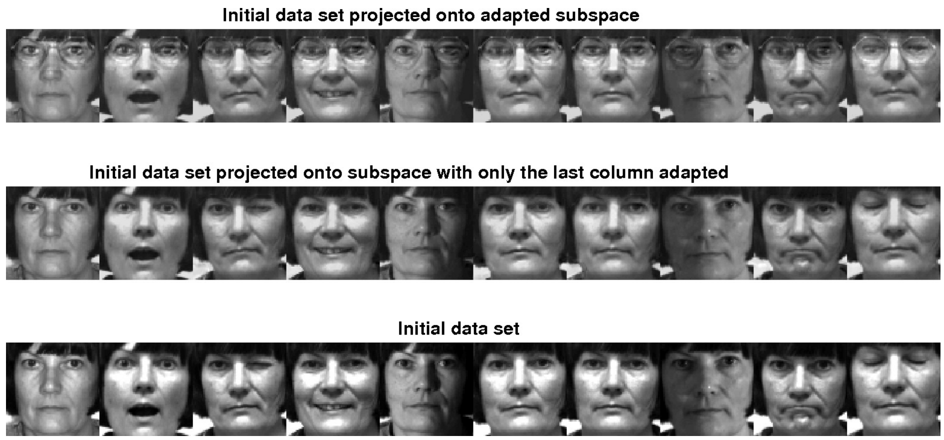


Fig. 7. *Initial face data set (bottom row) and its projection onto the corresponding POD subspace with only the last column adapted to the training excerpt (middle row) and its projection onto the fully-adapted POD subspace (top row).*

production of the picture excerpt but also makes use of the information that was previously sampled. We repeat the exercise with modifying only the last column of the initial POD subspace representative U_0 according to Corollary 10.

The gappy POD approximations using the adapted subspaces are shown in the last two columns of Fig. 6, again in comparison with the projection of the reference image onto the respective subspace. As is clear from Corollary 4,[6]

The important thing is how the adapted subspaces have changed. This can be visualized by projecting the initial snapshot ensemble onto the adapted subspaces, see Fig. 7. Apart from the fact that the bright white spots in the original data set are reproduced in a graying way when projected onto the last-column adapted subspace, these two data sets look almost the same (Fig. 7, bottom rows). In contrast, the original data set projected onto the fully adapted subspace features the property 'glasses-on' throughout (Fig. 7, top row). Nevertheless, the subspace distance between $[U_0]$ and the fully adapted $[U^*]$ is 0.1273, while the distcance between $[U_0]$ and the subspace $[U^*]$ with only the last column adjusted is 1.2828, more than ten times as large. Recall from Remark 7 that the latter $[U^*]$ corresponds to an SVD update with respect to a column-replacement in the original subspace representative U_0.

Additional experiments are featured in Section S3 from the supplement. The supplement also includes MATLAB code for the adapted gappy POD examples discussed here.

5. Summary and conclusion.

Subspace update problems arise in model reduction, machine learning, pattern recognition and computer vision. This paper focuses on the particular use case of subspace adaptation in combination with the model reduction methods of gappy POD and DEIM. These methods have in common that a mask matrix is utilized to extract the features deemed most important to the underlying problem. In both cases, the objective of the downstream subspace adaptation is to produce subspaces that contain elements that match the selected components. We

[6] which transfers in an analogous form to the sub-subspace setting of Subsection 3.4 both reconstructed images coincide since they both correspond to copying the training set to the respective entries of the unmodified gappy POD solution.

have formalized this objective as a nonlinear equation on the Grassmann manifold and have provided a closed-form solution that builds on the GROUSE approach [7, 49].

In the DEIM test case, discussed in Section 4.1.4, the mask matrix operates on vectors contained in the subspace that represents the nonlinear terms of the underlying discretized PDE. In the gappy POD test cases, discussed in Section 4.2, the mask matrix selects the important components from vectors contained in the subspace of state vector solution candidates. In the test case of DEIM-based model reduction, the Grassmann subspace update is used as an online adaptation method to improve the fit of the components sampled from the nonlinear term. The reduced model with online subspace updating achieves an average error of about one order of magnitude lower than a classical reduced model without the adaptation. In the gappy POD image processing example, the Grassmann subspace update is applied to implement a new feature in the subspace of solution candidates that is not contained in the sample data set. We expect the method to show similar advantages when used in combination with the missing point estimation [6], because of the similarities to DEIM and gappy POD.

Acknowledgments. The authors would like to thank Laura Balzano (University of Michigan) for helpful conversations concerning GROUSE. Moreover, the authors thank the reviewers and Associate Editor for inspiring comments that improved this manuscript. The first author was funded via a fellowship awarded by the German Research Foundation (DFG), grant number Zi 1250/2-1. The second and third authors acknowledge support of the U.S. Department of Energy, Awards DE-FG02-08ER2585 and DE-SC0009297, as part of the DiaMonD Multifaceted Mathematics Integrated Capability Center, and the AFOSR MURI grant FA9550-15-1-0038.

Appendix A. A direct solution of the Grassmann residual equation (11). This appendix features a short solution of (11). Obviously, (11) is solved, if we can find $t^* \in \mathbb{R}$ and $\alpha^* \in \mathbb{R}^p$ such that $P^T U(t^*)\alpha^* = b$, where $U(t^*) := U_0 + \left((\cos(t^* s_1) - 1)U_0 v + \sin(t^* s_1)\frac{Pr}{\|r\|}\right) v^T$. (All occurring quantities to be understood as introduced in Theorem 3.) Mind that $v = \alpha/\|\alpha\|_2$. Using an additional real parameter λ and the ansatz $\alpha^* = \lambda\alpha = \lambda v \|\alpha\|_2$ leads to the equation

$$\lambda \cos(t^* s_1)\left(\left(1 - \tan(t^* s_1)\frac{\|\alpha\|_2}{\|r\|_2}\right) P^T U_0 \alpha + \tan(t^* s_1)\frac{\|\alpha\|_2}{\|r\|_2} b\right) = b. \tag{34}$$

By setting $t^* = \frac{1}{s_1}\arctan\left(\frac{\|r\|_2}{\|\alpha\|_2}\right)$, the terms involving $P^T U_0$ cancel which leaves an equation for λ:

$$\lambda \cos(\arctan\left(\frac{\|r\|_2}{\|\alpha\|_2}\right))b = b.$$

The solution is $\lambda = \frac{1}{\cos(\arctan\left(\frac{\|r\|_2}{\|\alpha\|_2}\right)} = \sqrt{1 + \frac{\|r\|_2^2}{\|\alpha\|_2^2}}$.

In addition to its concision, this approach has the advantage that it simultaneously gives both t^* and the associated vector of coefficients $\alpha^* = \sqrt{(\frac{\|r\|_2}{\|\alpha\|_2} + 1)}\alpha \in \mathbb{R}^p$. On the other hand it does not allow to keep track of the residual depending on t, because for $t \neq t^*$, a defining equation is missing and $\alpha(t)$ and α are not collinear.

Nevertheless, we remark that the above short cut approach may be adapted to apply also in the setting of Corollary 10 from Subsection 3.4. In this case, one can work from the ansatz $\alpha^* = (\alpha_1, \ldots, \alpha_{p-l}, \lambda(\alpha_{p-l+1}, \ldots, \alpha_p))^T$.

One may also start by first applying the orthogonal coordinate transformation $\Phi = (v, Z) \in O_p$ to the subspace representative U_0, where $Z \in \mathbb{R}^{p\times(p-1)}$ contains an arbitrary orthonormal basis of $v^\perp$, and then work with $U_0\Phi$, $U(t^*)\Phi$. This course of action essentially leads to (34) appearing in the first column of $U(t^*)\Phi$ and the rest of the argument is analogous. See [49, App. C, Proof of Lemma 4] for related considerations.

Appendix B. Addendum to Subsection 3.4. A simple example of a differentiable Grassmann objective function for which Proposition 9 does not hold is

$$f : Gr(n,p) \to \mathbb{R}, \quad [U] \mapsto x^T U U^T y,$$

where $x, y \in \mathbb{R}^n$ are not orthogonal to $[U]$.

By using the basic fact that $D_X(v^T X w) = \left(\frac{\partial}{\partial x_{ij}} v^T X w\right)_{ij} = vw^T$ and the product rule, we see that the Grassmann gradient is

$$\nabla_{[U]} f = (I - UU^T) D_U f = (I - UU^T)\left(xy^T + yx^T\right) U,$$

where $D_U f = \left(\frac{\partial f}{\partial u_{i,j}}\right)_{i,j} \in \mathbb{R}^{n\times p}$, see [25, eq. (2.70)]. (Note that $\nabla_{[U]} f$ is of rank two in general, but of rank one, if $x = y$.) Introducing $U = (U_1, U_2)$ with $U_1 \in St(n, p-l)$, $U_2 \in St(n, l)$, we may write $UU^T = U_1U_1^T + U_2U_2^T$. By fixing U_1, f becomes a function $f_2 : Gr(n,l) \to \mathbb{R}, [U_2] \mapsto x^T U_1 U_1^T y + x^T U_2 U_2^T y$. The gradient is

$$\nabla_{[U_2]} f_2 = (I - U_2U_2^T)\left(xy^T + yx^T\right) U_2 \in \mathbb{R}^{n\times l}.$$

Likewise, for $f_1 : Gr(n, p-l) \to \mathbb{R}, [U_1] \mapsto x^T U_1 U_1^T y + x^T U_2 U_2^T y$, we obtain

$$\nabla_{[U_1]} f_1 = (I - U_1U_1^T)\left(xy^T + yx^T\right) U_1 \in \mathbb{R}^{n\times l}.$$

Splitting up the original gradient into an $(n \times (p-l))$ and an $(n \times l)$ matrix gives

$$\begin{aligned}
\nabla_{[U]} f &= \left((I - UU^T)(xy^T + yx^T)U_1, (I - UU^T)(xy^T + yx^T)U_2\right) \\
&\neq \left((I - U_1U_1^T)(xy^T + yx^T)U_1, (I - U_2U_2^T)(xy^T + yx^T)U_2\right) \\
&= \left(\nabla_{[U_1]} f_1, \nabla_{[U_2]} f_2\right).
\end{aligned}$$

In particular, $U_1^T \nabla_{[U_2]} f_2 = U_1^T xy^T U_2 + U_1^T yx^T U_2 \neq 0$ and the geodesic $U_2(t)$ in $Gr(n,l)$ along the gradient direction $\nabla_{[U_2]} f_2$ is not orthogonal to U_1,

$$U_1^T U_2(t) \neq 0.$$

A sufficient condition for (25) and Proposition 9 to hold is $(I - UU^T) D_U f = D_U f$ or, in short, $U^T D_U f = 0$.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Riemannian geometry of Grassmann manifolds with a view on algorithmic computation*, Acta Applicandae Mathematica, 80 (2004), pp. 199–220.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, New Jersey, 2008.

[3] D. Amsallem and C. Farhat, *Interpolation method for adapting reduced-order models and application to aeroelasticity*, AIAA Journal, 46 (2008), pp. 1803–1813.

[4] D. Amsallem and C. Farhat, *An online method for interpolating linear parametric reduced-order models*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2169–2198.

[5] D. Amsallem, M. Zahr, and C. Farhat, *Nonlinear model order reduction based on local reduced-order bases*, International Journal for Numerical Methods in Engineering, 92 (2012), pp. 891–916.

[6] P. Astrid, S. Weiland, K. Willcox, and T. Backx, *Missing points estimation in models described by proper orthogonal decomposition*, IEEE Transactions on Automatic Control, 53 (2008), pp. 2237–2251.

[7] L. Balzano, R. Nowak, and B. Recht, *Online identification and tracking of subspaces from highly incomplete information*, in Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on, IEEE, 2010, pp. 704–711.

[8] L. Balzano and S. J. Wright, *On GROUSE and incremental SVD*, in 2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013, pp. 1–4.

[9] L. Balzano and S. J. Wright, *Local convergence of an algorithm for subspace identification from partial data*, Foundations of Computational Mathematics, 15 (2015), pp. 1279–1314.

[10] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations*, Comptes Rendus Mathematique, 339 (2004), pp. 667–672.

[11] E. Begelfor and M. Werman, *Affine invariance revisited*, 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2 (2006), pp. 2087–2094.

[12] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, *Eigenfaces vs. fisherfaces: Recognition using class specific linear projection*, IEEE Trans. on PAMI, 19 (1997), pp. 711–720.

[13] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Review, 57 (2015), pp. 483–531.

[14] N. Boumal and P.-A. Absil, *Low-rank matrix completion via preconditioned optimization on the Grassmann manifold*, Linear Algebra and its Applications, 475 (2015), pp. 200–239.

[15] M. Brand, *Incremental Singular Value Decomposition of Uncertain Data with Missing Values*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 707–720.

[16] M. Brand, *Fast low-rank modifications of the thin singular value decomposition*, Linear Algebra and its Applications, 415 (2006), pp. 20–30.

[17] T. Bui-Thanh, M. Damodaran, and K. Willcox, *Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics*, AIAA Journal, 42 (2004), pp. 1505–1516.

[18] J. R. Bunch and C. P. Nielsen, *Updating the singular value decomposition*, Numerische Mathematik, 31 (1978), pp. 111–129.

[19] D. Cai, X. He, J. Han, and H.-J. Zhang, *Orthogonal laplacianfaces for face recognition*, IEEE Transactions on Image Processing, 15 (2006), pp. 3608–3614.

[20] K. Carlberg, *Adaptive h-refinement for reduced-order models*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 1192–1210. Special issue on model reduction.

[21] S. Chaturantabut and D. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2737–2764.

[22] A. D., M. Zahr, and K. Washabaugh, *Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction*, Special issue on Model Reduction of Parameterized Systems (MoRePaS), Advances in Computational Mathematics, (2014). (in review).

[23] J. Degroote, J. Vierendeels, and K. Willcox, *Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis*, International Journal for Numerical Methods in Fluids, 63 (2010), pp. 207–230.

[24] M. Dihlmann, M. Drohmann, and B. Haasdonk, *Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning*, in Proceedings of the International Conference on Adaptive Modeling and Simulation, D. Aubry, P. Dez, B. Tie, and N. Pars, eds., 2011, pp. 156–167.

[25] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM Journal on Matrix Analysis and Application, 20 (1999), pp. 303–353.

[26] J. Eftang and B. Stamm, *Parameter multi-domain hp empirical interpolation*, International Journal for Numerical Methods in Engineering, 90 (2012), pp. 412–428.

[27] R. Everson and L. Sirovich, *Karhunen-Loève procedure for gappy data*, Journal of the Optical Society of America, 12 (1995), pp. 1657–1664.

[28] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas, *Non-linear model reduction for uncertainty quantification in large-scale inverse problems*, International Journal for

Numerical Methods in Engineering, 81 (2010), pp. 1581–1608.
[29] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore – London, 3 ed., 1996.
[30] S. Kaulmann and B. Haasdonk, *Online greedy reduced basis construction using dictionaries*, in Proceedings of 7th Vienna International Conference on Mathematical Modelling, I. Troch and F. Breitenecker, eds., 2012, pp. 112–117.
[31] R. Kennedy, L. Balzano, S. J. Wright, and C. J. Taylor, *Online algorithms for factorization-based structure from motion*, Comput. Vis. Image Underst., 150 (2016), pp. 139–152.
[32] S. Kobayashi and K. Nomizu, *Foundations of Differential Geometry*, vol. I of Interscience Tracts in Pure and Applied Mathematics no. 15, John Wiley & Sons, New York – London – Sidney, 1963.
[33] J. Lee, *Riemannian Manifolds: an Introduction to Curvature*, Springer Verlag, New York – Berlin – Heidelberg, 1997.
[34] Y. Maday and B. Stamm, *Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2417–A2441.
[35] Y. Man Lui, *Advances in matrix manifolds for computer vision*, Image and Vision Computing, 30 (2012), pp. 380–388.
[36] T. S. Nguyen, *A real time procedure for affinely dependent parametric model order reduction using interpolation on Grassmann manifolds*, International Journal for Numerical Methods in Engineering, online (2012).
[37] M. Ohlberger and F. Schindler, *Error control for the localized reduced basis multi-scale method with adaptive on-line enrichment*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2865–A2895.
[38] H. Panzer, J. Mohring, R. Eid, and B. Lohmann, *Parametric model order reduction by matrix interpolation*, Automatisierungstechnik, 58 (2010), pp. 475–484.
[39] A. Paul-Dubois-Taine and D. Amsallem, *An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models*, International Journal for Numerical Methods in Engineering, 102 (2014), pp. 1262–1292.
[40] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz, *Localized discrete empirical interpolation method*, SIAM Journal on Scientific Computing, 36 (2014), pp. A168–A192.
[41] B. Peherstorfer and K. Willcox, *Detecting and adapting to parameter changes for reduced models of dynamic data-driven application systems*, in International Conference on Computational Science, vol. 51 of Procedia Computer Science, Elsevier, Elsevier, 2015, pp. 2553 – 2562.
[42] B. Peherstorfer and K. Willcox, *Dynamic data-driven reduced-order models*, Computer Methods in Applied Mechanics and Engineering, 291 (2015), pp. 21 – 41.
[43] B. Peherstorfer and K. Willcox, *Online adaptive model reduction for nonlinear systems via low-rank updates*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2123–A2150.
[44] L. Peng and K. Mohseni, *An online manifold learning approach for model reduction of dynamical systems*, SIAM Journal on Numerical Analysis, 52 (2014), pp. 1928–1952.
[45] M.-L. Rapùn and J. Vega, *Reduced order models based on local POD plus Galerkin projection*, Journal of Computational Physics, 229 (2010), pp. 3046–3063.
[46] K. Washabaugh, D. Amsallem, M. Zahr, and C. Farhat, *Nonlinear model reduction for CFD problems using local reduced-order bases*, in 42nd AIAA Fluid Dynamics Conference and Exhibit, Fluid Dynamics and co-located Conferences, AIAA Paper 2012-2686, AIAA, 2012, pp. 1–16.
[47] Y.-C. Wong, *Differential geometry of Grassmann manifolds*, Proceedings of the National Academy of Sciences of the United States of America, 57 (1967), pp. 589–594.
[48] D. Zhang and L. Balzano, *Convergence of a grassmannian gradient descent algorithm for subspace estimation from undersampled data*, CoRR, abs/1610.00199 (2016), http://arxiv.org/abs/1610.00199.
[49] D. Zhang and L. Balzano, *Global Convergence of a Grassmannian Gradient Descent Algorithm for Subspace Estimation*, Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS, Cadiz, Spain, May 2016, pp. 1460–1468.
[50] R. Zimmermann, *A locally parametrized reduced order model for the linear frequency domain approach to time-accurate computational fluid dynamics*, SIAM Journal on Scientific Computing, 36 (2014), pp. B508–B537.

SUPPLEMENTARY MATERIALS: GEOMETRIC SUBSPACE UPDATES WITH APPLICATIONS TO ONLINE ADAPTIVE NONLINEAR MODEL REDUCTION*

RALF ZIMMERMANN†, BENJAMIN PEHERSTORFER‡, AND KAREN WILLCOX§

S1. Supplementary example to Subsection 2.2. This section illustrates the basic objective introduced in Subsection 2.2 via an example in $Gr(3,1)$. Points on $Gr(3,1)$ are represented by orthogonal (3×1)-matrices, i.e., vectors on the unit sphere $S^2 = \{(x,y,z)^T \in \mathbb{R}^3 | \quad x^2+y^2+z^2 = 1\}$, and can thus be conveniently visualized. Suppose that a starting subspace $\mathcal{U}_0 = [u_0] \in Gr(3,1) \cong S^2$ is given, where $u_0 \in S^2$. Suppose further that target data for the x and y coordinates are specified, say, $x = b_1, y = b_2$. We are looking for a subspace $\mathcal{U}^* = [u^*], u^* \in S^2$ that contains vectors that match the target data:

(S1)

$$[u^*] \in \mathcal{Z} := \{[u] \in Gr(3,1) | \min_{\alpha\in\mathbb{R}} \|P^T u\alpha - b\|^2 = 0\}, \quad P^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

The set $\mathcal{Z}$ contains infinitely many global solutions to the least-squares optimization problem. Any unit-2-norm vector $u \in S^2$ whose first two components are in the span of the target vector b, $(u_1,u_2)^T = \lambda(b_1,b_2)^T$ represents a global optimum. Hence, the set of global optima is

$$\mathcal{Z} = \left\{ \begin{pmatrix} \lambda b_1 \\ \lambda b_2 \\ \sqrt{1-\lambda^2\|b\|_2^2} \end{pmatrix} | \quad \lambda \in \left[-\frac{1}{\|b\|_2}, \frac{1}{\|b\|_2}\right] \setminus \{0\} \right\}.$$

This corresponds to (10). For example, two 'easy-to-construct' trivial solutions are

$$u_{tr1} := \frac{(b_1,b_2,0)^T}{\|(b_1,b_2,0)^T\|_2}, \quad u_{tr2} := \frac{(b_1,b_2,u_{0_3})^T}{\|(b_1,b_2,u_{0_3})^T\|_2},$$

i.e., we simply take the target data and fill up with zeros ('tr1') or we copy the target data to the x and y coordinates of the starting point u_0 and renormalize ('tr2').

In the academic case at hand, it is straightforward to compute the minimizer to the following nonlinear *constrained* Grassmann optimization problem

$$\text{(S2)} \qquad [z^*] := \operatorname*{arg\,min}_{[u]\in Gr(3,1)} \operatorname{dist}([u_0],[u]), \quad \text{s.t. } [u] \in \mathcal{Z},$$

which is given by

$$z^* := \mathcal{Z}(\lambda^*), \quad \lambda^* = \pm\frac{|\langle b, P^T u_0\rangle|}{\sqrt{u_{0_3}^2\|b\|^2 + \langle b, P^T u_0\rangle^2\|b\|}}.$$

*The first author was funded via a research fellowship awarded by the German Research Foundation (DFG), grant number Zi 1250/2-1. The second and third authors acknowledge support of the U.S. Department of Energy, Awards DE-FG02-08ER2585 and DE-SC0009297, as part of the DiaMonD Multifaceted Mathematics Integrated Capability Center.

†Department of Mathematics and Computer Science, SDU Odense, Denmark and Department of Aeronautics & Astronautics, MIT, Cambridge, USA (zimmermann@imada.sdu.dk).

‡Department of Mechanical Engineering, University of Wisconsin-Madison, USA (peherstorfer@wisc.edu).

§Department of Aeronautics & Astronautics, MIT, Cambridge, USA kwillcox@mit.edu).

(The sign of λ^* depends on the sign of $\langle b, P^T u_0\rangle$.)

We may construct another solution via following a shortest path along the negative of the gradient of the Grassmann function F from (5) that is associated with (S1). It turns out that this path crosses the feasibility set $\mathcal{Z}$. We denote the resulting solution by $[u^*] \in Gr(3,1)$. In Section 3, we derive a closed formula for computing such subspaces $[U^*]$ on Grassmann manifolds of arbitrary dimension.

Figure S1 displays the base point u_0, the set of global least-squares optima $\mathcal{Z}$, the exact optimum $z^* = \mathcal{Z}(\lambda^*)$ of the constrained Grassmann problem (S2) as well as u^*, u_{tr1}, u_{tr2}, where

$$u_0 = \begin{pmatrix} 0.6548 \\ 0.3706 \\ 0.6587 \end{pmatrix}, b = \begin{pmatrix} 0.7046 \\ 0.6601 \end{pmatrix}.$$

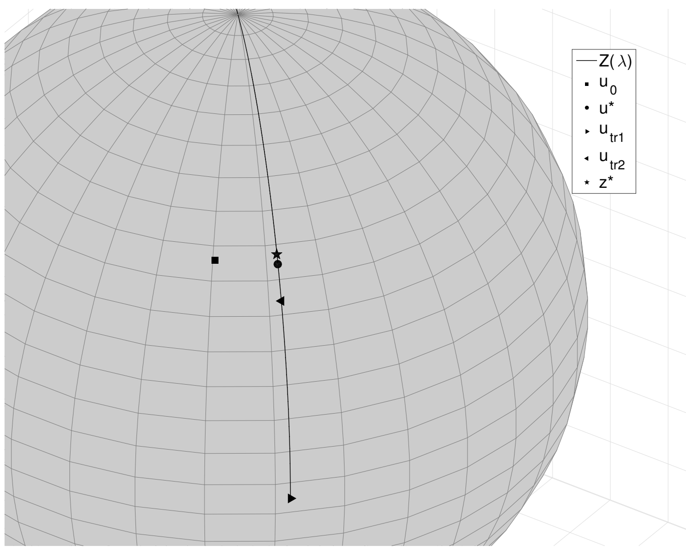


FIG. S1. *Reference point u_0 and the set of least-squares optimal solutions $\mathcal{Z} = \{\mathcal{Z}(\lambda)\}$ to the problem (S1). The associated subspaces are spanned by the vectors pointing to the curve $\mathcal{Z}$. On $\mathcal{Z}$ lie the arbitrary 'trivial' solutions u_{tr1}, u_{tr2} as well as the optimal solution $\mathcal{Z}(\lambda^*)$ to (S2) and the solution u^* obtained by following a shortest path along the negative of the gradient associated with (S1) starting in u_0.*

S2. Additional comments on the connection of GROUSE and rank-one SVD updates. In Remark 7 and Remark 11, we have briefly commented on the connection between the residual-annihilating GROUSE t^*-update and the SVD update procedures of [S3, S2] and [S1]. In particular, it was claimed in Remark 11 that the 'revise'-method of [S2, Table 1] and the GROUSE t^*-update restricted to the last column of a given initial subspace representative U_0 coincide. Here is the proof:

Written column-wise, $U_0 = (u_0^1, \dots, u_0^{p-1} \,|u_0^p)$. When using the method of Subsection 3.4 applied to the last column u_0^p, we arrive at

$$U^* = (u_0^1, \dots, u_0^{p-1} \,|u_0^p(t^*)).$$

According to Corollary 10,

$$u_0^p(t^*) = \frac{1}{\sqrt{\|r\|_2^2 + \alpha_p^2}} \left(u_0^p \alpha_p + Pr\right) = \frac{(u_0^p \alpha_p + Pr)}{\|\left(u_0^p \alpha_p + Pr\right)\|_2}, \tag{S3}$$

where P is the mask matrix and $r = b - P^T U_0 \alpha$, $\alpha = (\alpha_1 \ldots, \alpha_p)^T$.

The 'revise'-method of [S2, Table 1] proceeds as follows: In our setting, $U_0 = (u_0^1, \ldots, u_0^{p-1} \,|u_0^p) \in \mathbb{R}^{n\times p}$ plays the role of $[X, c]$ from [S2, Table 1]. The objective is to replace the column $c = u_0^p$ with a new column $d = U_0\alpha + Pr$. Obviously $P^T d = P^T U_0 \alpha + P^T P r = P^T U_0 \alpha + r = P^T U_0 \alpha + (b - P^T U_0 \alpha) = b$. This means that a subspace that contains this direction d is in the 'feasibility set' $\mathcal{Z}$ introduced in (10).

In [S2], the column exchange is rewritten as a rank-one update of the following form:

$$U' S' V'^T = U S V^T + ab^T \overset{\text{here}}{=} U_0 + ab^T, \quad a = d - c, \quad b^T = e_p^T = (0, \ldots, 0, 1).$$

The matrix $U_0 + ab^T$ is precisely the matrix U_0 with the last column replaced by d, i.e.,

$$U_0 + ab^T = (U_0^{p-1} \,|d) = (U_0^{p-1} \,|U_0\alpha + Pr).$$

In particular, for the U'-factor in the revised SVD:

$$\operatorname{colspan}(U') = \operatorname{colspan}((U_0^{p-1} \,|U_0\alpha + Pr)) = \operatorname{colspan}(\left(U_0^{p-1} \,| \frac{\Pi_\perp^{p-1}(U_0\alpha + Pr)}{\|\Pi_\perp^{p-1}(U_0\alpha + Pr)\|}\right)) \tag{S4}$$

where $\Pi_\perp^{p-1} = (I - U_0^{p-1}(U_0^{p-1})^T)$ is the orthogonal projection onto the orthogonal complement of $\operatorname{colspan}(U_0^{p-1})$. This is just the Gram-Schmidt step. Note that $\Pi_\perp^{p-1}(U_0\alpha + Pr) = u_0^p\alpha_p + Pr$, so that the last column of (S4) indeed coincides with the last column of (S3). All additional operations like subspace rotations that are inherent in the procedure of [S2] do not affect the column-span.

In order to comment on the connection to [S1], we go into full detail. The method of [S2] starts with a detour via $p+1$ columns in the factorization

$$U_0 + ab^T = (U_0, q) \left(\begin{array}{c|c} I_p & U_0^T a \\ 0 & \|\tilde{q}\|_2 \end{array}\right) \begin{pmatrix} I_p \\ e_p^T \end{pmatrix}, \quad q = \frac{\tilde{q}}{\|\tilde{q}\|}, \quad \tilde{q} = (I - UU^T)a. \tag{S5}$$

The above matrix product reduces to

$$U_0 + ab^T = \underbrace{(U_0, q)}_{n\times(p+1)} \underbrace{\begin{pmatrix} I_p + U_0^T a e_p^T \\ \hline 0, \ldots, 0, \|\tilde{q}\|_2 \end{pmatrix}}_{(p+1)\times p} = (U_0, q) \left(\begin{array}{c|c} I_{p-1} & (U_0^T a)_1^{p-1} \\ \hline 0, \ldots, 0 & \langle u_0^p, a\rangle + 1 \\ \hline 0, \ldots, 0 & \|\tilde{q}\|_2 \end{array}\right) =: (U_0, q) M.$$

Note that $U_0^T a = U_0^T (d - c) = U_0^T (U_0\alpha + Pr - u_0^p) = \alpha - e_p$. Thus,

$$M = \left(\begin{array}{c|c} & \alpha_1 \\ I_{p-1} & \vdots \\ & \alpha_{p-1} \\ \hline 0, \ldots, 0 & \alpha_p \\ \hline 0, \ldots, 0 & \|\tilde{q}\|_2 \end{array}\right). \tag{S6}$$

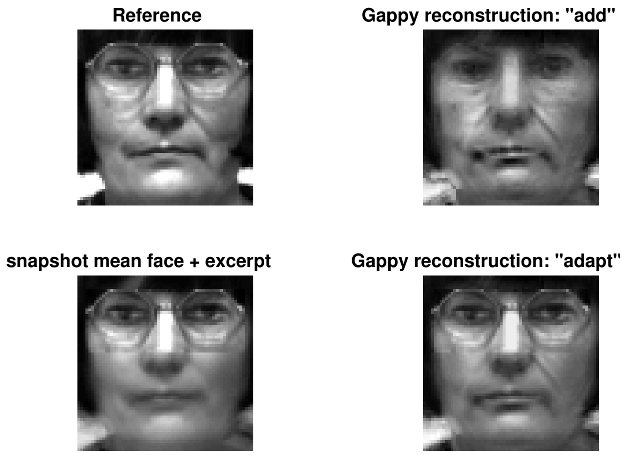


FIG. S2. *The geometric rank-one subspace adaptation in comparison with brute-force approaches to implant the picture excerpt into the subpace.*

The qr-decomposition of the matrix $M \in \mathbb{R}^{p+1\times p}$ is

$$M = QR = \underbrace{\left(\begin{array}{c|c} I_{p-1} & 0 \\ \hline 0,\dots,0 & x \\ \hline 0,\dots,0 & y \end{array}\right)}_{Q\in\mathbb{R}^{(p+1)\times p}} \underbrace{\left(\begin{array}{c|c} I_{p-1} & \alpha_1^{p-1} \\ \hline 0,\dots,0 & \nu \end{array}\right)}_{R\in\mathbb{R}^{p\times p}},$$

where $x = \frac{\alpha_p}{\nu}$, $y = \frac{\|\tilde{q}\|_2}{\nu}$, $\nu = \sqrt{\alpha_p^2 + \|\tilde{q}\|^2}$. As a consequence

$$(U,q)Q = \left(u_0^1,\dots,u_0^{p-1}\ |xu_0^p + yq\right) = \left(u_0^1,\dots,u_0^{p-1}\ |\frac{1}{\nu}(\alpha_p u_0^p + Pr)\right),$$

since $\tilde{q} = (I - U_0U_0^T)a = (I - U_0U_0^T)(U_0\alpha + Pr - u_0^p) = Pr$. This is precisely the same matrix representative as in (S4) and its last column equals (S3). Formally, [S2] requires to compute the SVD of M but this is equivalent to computing Q times the SVD of R. Up to a rotation, we obtain always the same 'subspace factor', as the theory predicts.

In [S1, Alg. 3], a similar decompositon $(U,q)\tilde{M}$ as in (S5) appears. The difference is that there, the matrix factor $\tilde{M}$ is a square $(p+1)\times(p+1)$-matrix,

$$\tilde{M} = \left(\begin{array}{c|c} I_p & \alpha \\ \hline 0,\dots,0 & \|\tilde{q}\|_2 \end{array}\right) \in \mathbb{R}^{(p+1)\times(p+1)}.$$

The matrix $M \in \mathbb{R}^{(p+1)\times p}$ in (S6) features the same last column but shifted to the left. While this corresponds to *replacing* data in the original subspace representative, the $\tilde{M}$ from [S1, Alg. 3] corresponds to *appending* data, which is followed by a truncation procedure.

S3. Additional results for the example of Subsection 4.2. In this section, we conduct two complementary experiments to the gappy POD image processing example of Subsection 4.2. We consider two brute-force approaches of adding the picture excerpt displayed in Figure 6, lower left corner, to the POD subspace formed from the face database displayed Figure 4.

The first approach is as follows: we start with the unprocessed snapshot matrix $Y = (y_1, \ldots, y_{10}) \in \mathbb{R}^{4096\times 10}$. Then, we compute the snapshot mean vector $y_{mean} = \frac{1}{10}\sum_{k=1}^{10} y_k$ and replace the entries $P^T y_{mean}$ with those of the picture excerpt, i.e., we construct $y_{add} \in \mathbb{R}^{4096}$ such that $P^T y_{add} = P^T y^g$ where y^g is the gappy data vector. The remaining entries of y_{add} coincide with those of the mean vector. We add y_{add} to the snapshot matrix, recompute the SVD and truncate to the original dimension of 10 basis vectors:

$$U_{add}\Sigma_{add}V_{add}^T \overset{\text{SVD}}{=} (Y, y_{add}) \in \mathbb{R}^{4096\times 11}, \quad U_{add} := (u_{add}^1, \ldots, u_{add}^{10}) \in St(4096, 10).$$

The best gappy POD reconstruction that is based on the subspace $[U_{add}]$ is shown in Figure S2 in the upper right corner.

The subspace distance between the initial POD space $[U_0]$ and $[U_{add}]$ is

$$\operatorname{dist}([U_0], [U_{add}]) = 0.13525.$$

The second approach works by *replacing* the last column of the POD subspace representative U_0 with the artificially constructed vector y_{add} followed by recomputing the SVD:

$$U_{rep}\Sigma_{rep}V_{rep}^T \overset{\text{SVD}}{=} (u_0^1, \ldots, u_0^9, y_{add}) \in \mathbb{R}^{4096\times 10}, \quad U_{rep} \in St(4096, 10).$$

Since the subspace U_{rep} now contains the vector y_{add}, the associated gappy POD reconstruction coincides with y_{add} and thus looks the same as Figure S2 in the lower left corner. The subspace distance between the initial POD space $[U_0]$ and $[U_{rep}]$ is

$$\operatorname{dist}([U_0], [U_{rep}]) = 1.5685.$$

The subspace distance between the initial POD space $[U_0]$ and the geometric rank-one update $[U^*]$ from Section 3 is

$$\operatorname{dist}([U_0], [U^*]) = 0.12734.$$

This confirms that the adapted subspace $[U^*]$ is closer to the initial POD subspace than its competitors. Moreover, it is even cheaper to obtain, since it avoids an extra SVD. Theoretically, it corresponds to inputing the vector $U_0\alpha + Pr$ after a suitable rotation of the subspace representative U_0. The brute-force approach of adding an artificial snapshot to the database and doing the POD from scratch does not lead to a satisfactory result. The brute-force approach of replacing a column of the initial POD basis matrix with the artificial snapshot leads to a much larger gap in the subspace distance.

This supplement includes MATLAB code for the above example.

S4. MATLAB code for the geometric rank-one update. The following MATLAB code corresponds to Corollary 4 and Corollary 10.

```
%
% file Grassmann_res_update_masked.m
%
function [U, PTU] =...
```

```
    Grassmann_res_update_masked(U0, P, b, lastcols)
%--------------------------------------------------------------
% Grassmann_res_update_masked
%   compute root of the residual function res: G(n,p) -> R
%   corrsponding to a masked least-squares system
%             min||P'Ux - P'b||
%   on the Grassmann-Manifold G(n,p)
%
% input arguments
%         U0 : orthogonal representative of point in G(n,p)
%          P : list of slected points
%          b : right hand side, filtered by
%              the mask operator, i.e. b(P,:)
%   lastcols : number of columns to be adapted,
%              counted from rear:
%              e.g. lastcols = 4 means that subspace
%              representative U in R^(n x p)
%              is decomposed into
%                   U = (U(:,1:p-4), U(:, p-4+1:p))
%              and only the subspace spanned
%              by the last 4 columns is adapted
%              lastcols = 0 means: adapt FULL subspace
% @Output:
%          U : adapted subspace representative
%        PTU : P'*U = U(P,:)
%
% author: R: Zimmermann, IMADA, SDU Odense
%   zimmermann@imada.sdu.dk
%--------------------------------------------------------------

% produce onscreen output?
onscreen = 0;

% get dimensions
[n, p] = size(U0);
if lastcols == p
    lastcols = 0;
end

%--------------------------------------------------------------
% Closed form solution:
% ********************
%
% The gradient is the rank-one matrix
%   G = -2P(b-P^TU alpha)*alpha^T
%     = -2P(b-QQ^Tb)*alpha^T.
% For the geodesic path that features H=-G as a starting
% velocity, we need the SVD of H. Since H is rank-one, it
% holds
%   svd(H) = (P*q) * sigma * v^T, where
```

```
%     q = r/|r|,
% sigma = 2*|r|*|alpha|,
%     v = alpha/|alpha|
%
% The geodesic is
%  U(t) = U0 + {(cos(t*sigma)-1)*U0*v + sin(t*sigma)*(P*q)}*v^T
%       = U0 + x(t)*v^T
%
% The optimal t is:   t_star = (1/sigma)*atan(|r|/|alpha|)
%------------------------------------------------------------

% compute vector of optimal coefficients and residual
% thin SVD
[Q,S,R] = svd(U0(P,:), 0);
% inverse of singular value matrices, stored as vector
S_inv = 1.0./diag(S);
QTb = Q'*b;
% compute vector of optimal coefficients
alpha = R*(S_inv.*(QTb));
%compute residual vector
r = b - Q*QTb;
n_r = norm(r);

if lastcols
    % keep only the components associated with the last cols
    alpha = alpha(p-lastcols+1:p);
end
n_alpha = norm(alpha);
v = alpha/n_alpha;
% optimal step
t_star = atan(n_r/n_alpha);

if lastcols == 0
    % Geodesic
    x = (cos(t_star)-1)*U0*v;
    x(P) = x(P) + (sin(t_star)/n_r)*r;
    U = U0 + x*v';
    %--------------------------------------------------------
    % compute projection after rank-1-update
    % in closed form
    %
    % The result is the same as recomputing the SVD of PTU:
    %  [Qopt, Sopt, Ropt] = svd(PTU, 0);
    %  residual_t_star = norm(b - Qopt*(Qopt'*b))
    %
    % Actually, it is not necessary to compute the residual
    % since it is theoretically guaranteed to be zero.
    % This is merely a check for the numerical accuracy.
    %--------------------------------------------------------
    if onscreen
```

```
        gp = -1.0/n_alpha*(S_inv.*S_inv).*QTb;
        gp1 = n_alpha/n_r;
        g = [gp,;gp1];
        n_g = norm(g);
        q = r/n_r;
        Qhat = [Q,q];
        Qhatg = (1./n_g)*Qhat*g;
        b_proj = Qhat*(Qhat'*b) - (Qhatg'*b)*Qhatg;
        check_residual = norm(b-b_proj)
        % For comparison: brute force via re-SVD
        [Qopt, Sopt, Ropt] = svd(U(P,:), 0);
        check_Lem2 = norm(b_proj - Qopt*(Qopt'*b))
    end
else
    % Geodesic
    x = (cos(t_star)-1)*U0(:,p-lastcols+1:p)*v;
    x(P) = x(P)+ (sin(t_star)/n_r)*r;
    % subspace update
    U = [U0(:,1:p-lastcols), U0(:,p-lastcols+1:p) + x*v'];
end

PTU = U(P,:);
return;
end
% end of file Grassmann_res_update_masked.m
```

REFERENCES

[S1] L. Balzano and S. J. Wright, *On GROUSE and incremental SVD*, in 2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013, pp. 1–4.

[S2] M. Brand, *Fast low-rank modifications of the thin singular value decomposition*, Linear Algebra and its Applications, 415 (2006), pp. 20–30.

[S3] J. R. Bunch and C. P. Nielsen, *Updating the singular value decomposition*, Numerische Mathematik, 31 (1978), pp. 111–129.