

SIMPLER GRASSMANNIAN OPTIMIZATION

ZEHUA LAI, LEK-HENG LIM, AND KE YE

ABSTRACT. There are two widely used models for the Grassmannian $\text{Gr}(k, n)$, as the set of equivalence classes of orthogonal matrices $O(n)/(O(k) \times O(n-k))$, and as the set of trace- k projection matrices $\{P \in \mathbb{R}^{n \times n} : P^\top = P = P^2, \text{tr}(P) = k\}$. The former, standard in manifold optimization, has the downside of relying on equivalence classes but working with orthogonal matrices is generally good numerical practice. The latter, widely adopted in coding theory and probability, uses actual matrices (as opposed to equivalence classes) but working with projection matrices is numerically unstable. We present an alternative that has both advantages and suffers from neither of the disadvantages; by representing k -dimensional subspaces as symmetric orthogonal matrices of trace $2k - n$, we obtain

$$\text{Gr}(k, n) \cong \{Q \in O(n) : Q^\top = Q, \text{tr}(Q) = 2k - n\}.$$

As with the other two models, we show that differential geometric objects and operations — tangent vector, metric, normal vector, exponential map, geodesic, parallel transport, gradient, Hessian, etc — have closed-form analytic expressions that are computable with standard numerical linear algebra. In the proposed model, these expressions are considerably simpler, a result of representing $\text{Gr}(k, n)$ as a linear section of a compact matrix Lie group $O(n)$, and can be computed with at most one QR decomposition and one exponential of a special skew-symmetric matrix that takes only $O(nk(n-k))$ time. In particular, we completely avoid eigen- and singular value decompositions in our steepest descent, conjugate gradient, quasi-Newton, and Newton methods for the Grassmannian. Another important feature of these algorithms, particularly evident in steepest descent and Newton method, is that they exhibit clear signs of numerical stability; various measures of errors consistently reduce to the order of machine precision throughout extensive numerical experiments.

1. INTRODUCTION

As a manifold, the Grassmannian $\text{Gr}(k, n)$ is just the set of k -planes in n -space with its usual differential structure; this is an abstract description that cannot be employed in algorithms and applications. In order to optimize functions $f : \text{Gr}(k, n) \rightarrow \mathbb{R}$ using currently available technology, one needs to put a coordinate system on $\text{Gr}(k, n)$. The best known way, as discovered by Edelman, Arias, and Smith in their classic work [24], is to realize $\text{Gr}(k, n)$ as a *matrix manifold* [2], where every point on $\text{Gr}(k, n)$ is represented by a matrix or an equivalence class of matrices and from which one may derive closed-form analytic expressions for other differential geometric objects (e.g., tangent, metric, geodesic) and differential geometric operations (e.g., exponential map, parallel transport) that in turn provide the necessary ingredients (e.g., Riemannian gradient and Hessian, conjugate direction, Newton step) for optimization algorithms. The biggest advantage afforded by the approach in [24] is that a judiciously chosen system of extrinsic *matrix coordinates* for points on $\text{Gr}(k, n)$ allows all aforementioned objects, operations, and algorithms to be computed solely in terms of standard *numerical linear algebra*, which provides a ready supply of stable and accurate algorithms [31] with high-quality software implementations [6]. In particular, one does not need to solve any differential equations numerically when doing optimization on matrix manifolds à la [24].

1.1. Existing models. There are two well-known models for $\text{Gr}(k, n)$ supplying such matrix coordinates — one uses orthogonal matrices and the other projection matrices. In optimization, the

2010 *Mathematics Subject Classification.* 14M15, 90C30, 90C53, 49Q12, 65F25, 62H12.

Key words and phrases. Grassmannian, Grassmann manifold, manifold optimization.

by-now standard model (see, for example, [23, 39, 51, 56, 65]) is the one introduced in [24], namely,

$$\mathrm{Gr}(k, n) \cong \mathrm{O}(n)/(\mathrm{O}(k) \times \mathrm{O}(n-k)) \cong \mathbb{V}(k, n)/\mathrm{O}(k), \quad (1)$$

where $\mathbb{V}(k, n) := \{V \in \mathbb{R}^{n \times k} : V^T V = I\} \cong \mathrm{O}(n)/\mathrm{O}(n-k)$ is the Stiefel manifold. In this homogeneous space model, which is also widely used in areas other than optimization [7, 8, 30, 32, 49, 50, 70], a point $\mathbb{V} \in \mathrm{Gr}(k, n)$, i.e., a k -dimensional subspace $\mathbb{V} \subseteq \mathbb{R}^n$, is represented by its orthonormal basis, written as columns of a matrix $V = [v_1, \dots, v_k] \in \mathbb{V}(k, n)$. Since any two orthonormal bases $V_1, V_2 \in \mathbb{V}(k, n)$ of \mathbb{V} must be related by $V_1 = V_2 Q$ for some $Q \in \mathrm{O}(k)$, such a representation is not unique and so this model requires that we represent \mathbb{V} not as a single $n \times k$ orthonormal matrix but as a whole equivalence class $[[V]] := \{VQ \in \mathbb{V}(k, n) : Q \in \mathrm{O}(k)\}$ of orthonormal bases of \mathbb{V} . A brief word about our notations: Throughout this article, we adopt the convention that a vector space $\mathbb{V} \in \mathrm{Gr}(k, n)$ will be typeset in blackboard bold, with the corresponding letter in normal typeface $V \in \mathbb{V}(k, n)$ denoting an (ordered) orthonormal basis. Equivalence classes will be denoted in double brackets, so $[[V]] = \mathbb{V}$. Diffeomorphism of two smooth manifolds will be denoted by \cong .

It is straightforward to represent a point $\mathbb{V} \in \mathrm{Gr}(k, n)$ by an actual matrix as opposed to an equivalence class of matrices. Since any subspace \mathbb{V} has a unique orthogonal projection matrix $P_{\mathbb{V}}$, this gives us an alternative model for the Grassmannian that is also widely used (notably in linear programming [58, 69] but also many other areas [15, 11, 17, 25, 48, 53]):

$$\mathrm{Gr}(k, n) \cong \{P \in \mathbb{R}^{n \times n} : P^T = P = P^2, \mathrm{tr}(P) = k\}. \quad (2)$$

Note that $\mathrm{rank}(P) = \mathrm{tr}(P) = \dim(\mathbb{V})$ for orthogonal projection matrices. The reader is reminded that an *orthogonal* projection matrix is not an orthogonal matrix — the ‘orthogonal’ describes the projection, not the matrix. To avoid confusion, we drop ‘orthogonal’ from future descriptions — all projection matrices in our article will be orthogonal projection matrices.

As demonstrated in [34], it is also possible to derive closed-form analytic expressions for various differential geometric objects and present various optimization algorithms in terms of the matrix coordinates in (2). Nevertheless, the problem with the model (2) is that algorithms based on projection matrices are almost always *numerically unstable*, especially in comparison with algorithms based on orthogonal matrices. This is likely the reason why there are no numerical experiments in [34]. Roughly speaking an orthogonal matrix preserves (Euclidean) norms and therefore rounding errors do not get magnified through a sequence of orthogonal transformations [21, Section 3.4.4] and consequently algorithms based on orthogonal matrices tend to be numerically stable (details are more subtle, see [66, pp. 124–166] and [36]). Projection matrices not only do not preserve norms but are singular and give notoriously unstable algorithms — possibly the best known illustration of numerical instability [63, 64] is one that contrasts Gram–Schmidt, which uses projection matrices, with Householder QR, which uses orthogonal matrices.¹ In fact, the proper way to compute projections is to do so via a sequence of orthogonal matrices [60, pp. 260–261], as a straightforward computation is numerically unstable [16, pp. 849–851].

The alternative (1) is currently universally adopted for optimization over a Grassmannian. One issue with the model (1) is that a point on $\mathrm{Gr}(k, n)$ is not a single matrix but an *equivalence class* of uncountably many matrices. Equivalence classes are tricky to implement in numerical algorithms and standard algorithms in numerical linear algebra [6] do not work with equivalence classes of matrices. Given a function $f : \mathrm{Gr}(k, n) \rightarrow \mathbb{R}$ to be optimized, any optimization algorithm [23, 24, 39, 51, 56, 65] that rely on the model (1) side steps the issue by lifting f to an $\mathrm{O}(k)$ -invariant function $\tilde{f} : \mathbb{V}(k, n) \rightarrow \mathbb{R}$, i.e., where $\tilde{f}(VQ) = \tilde{f}(V)$ for all $Q \in \mathrm{O}(k)$. This incurs additional costs in two ways: (a) whenever a point $\mathbb{V} \in \mathrm{Gr}(k, n)$ needs to be lifted to a point $V \in \mathbb{V}(k, n)$, this incurs the cost of finding an orthonormal basis V for \mathbb{V} ; (b) whenever one needs to check

¹For example, computing the QR decomposition of a Hilbert matrix $A = [1/(i+j-1)]_{i,j=1}^{15}$, we get $\|Q^*Q - I\| \approx 8.0 \times 10^0$ with Gram–Schmidt, 1.7×10^0 with modified Gram–Schmidt, 2.4×10^{-15} with Householder QR.

equality of points $\text{im}(V_1) \stackrel{?}{=} \text{im}(V_2)$, this incurs the cost of one matrix product $V_1^T V_2$ and its norm.² These additional costs cannot be avoided in the model (1) as we represent a linear subspace by an equivalence class. For comparison, (a) and (b) are immaterial when points are represented as actual matrices, like in model (2) or our proposed model. Moreover it is impossible to continuously choose such ‘Stiefel coordinates’ $V \in V(k, n)$ for every point $\mathbb{V} \in \text{Gr}(k, n)$, as we will discuss in Section 7.5. A second and more serious issue with the model (1) is that its associated optimization algorithms in [24] are still significantly less stable than those for our proposed model. As we will see in Section 8, and for reasons explained therein, loss-of-orthogonality remains very much a problem when we use (1) to represent a Grassmannian. This is likely the reason why the numerical experiments in [24] had used extended precision arithmetic.

We would like to mention a noncompact analogue of (1) that is popular in combinatorics [1, 26, 28, 44, 47]:

$$\text{Gr}(k, n) \cong \mathbb{R}_k^{n \times k} / \text{GL}(k), \quad (3)$$

where $\mathbb{R}_k^{n \times k} := \{A \in \mathbb{R}^{n \times k} : \text{rank}(A) = k\}$ and $\text{GL}(k) := \{X \in \mathbb{R}^{k \times k} : \det(X) \neq 0\}$. It has also been shown [2] that one may obtain closed-form analytic expressions for differential geometric quantities with the model (3) and so in principle one may use it for optimization purposes. Nevertheless, from the perspective of numerical algorithms, the model (3) suffers from the same problem as (2) — by working with rank- k matrices, i.e., whose condition number can be arbitrarily large, algorithms based on (3) are inherently numerically unstable. In fact, since the model (3) also represents points as equivalence classes, it has both shortcomings of (1) and (2) but neither of their good features. The natural redress of imposing orthogonal constraints on (3) to get a well-conditioned representative for each equivalence class would just lead one back to the model (1).

Looking beyond optimization, we stress that each of the aforementioned models has its own (sometimes unique) strengths. For example, (3) is the only model we know in which one may naturally define the positive Grassmannian [26], an important construction in combinatorics [44] and physics [28]. The model (2) is indispensable in probability and statistics as probability measures [48, Section 3.9] and probability densities [15, Section 2.3.2] on $\text{Gr}(k, n)$ are invariably expressed in terms of projection matrices.

1.2. Proposed model. We propose to use a model for the Grassmannian that combines the best features, suffers from none of the defects of the aforementioned models, and, somewhat surprisingly, is also simpler:

$$\text{Gr}(k, n) \cong \{Q \in \text{O}(n) : Q^T = Q, \text{tr}(Q) = 2k - n\}. \quad (4)$$

This model, which represents k -dimensional subspace as a symmetric orthogonal matrix of trace $2k - n$, is known but obscure. It was mentioned in passing in [10, p. 305] and was used in [41] to derive geodesics for the *oriented Grassmannian*, a different but related manifold. Note that (4) merely provides an expression for *points*, our main contribution is to derive expressions for other differential geometric objects and operations, as well as the corresponding optimization algorithms, thereby fully realizing (4) as a model for optimization. A summary of these objects, operations, and algorithms is given in Table 1. From a differential geometric perspective, Sections 2–5 may be regarded as an investigation into the embedded geometry of $\text{Gr}(k, n)$ as a submanifold of $\text{O}(n)$.

The two key advantages of the model (4) in computations are that:

- (i) we represent points on $\text{Gr}(k, n)$ as actual matrices, not equivalence classes;
- (ii) we work only with orthogonal matrices and in numerical stable ways.

Numerical stability is an important feature of the algorithms for model (4); as we will see in Section 8, the errors and gradients in our steepest descent and Newton algorithms consistently reduce to the order of machine precision. Moreover, another bonus with (4) is that the expressions

²Note that $\text{im}(V_1) \stackrel{?}{=} \text{im}(V_2)$ is equivalent to $V_1 V_1^T \stackrel{?}{=} V_2 V_2^T$, which is equivalent to $\|V_1^T V_2\| \stackrel{?}{=} k$ as $\|V_1 V_1^T - V_2 V_2^T\|^2 = 2k^2 - 2\|V_1^T V_2\|^2$. We thank the referee for this observation.

OBJECTS/OPERATIONS	RESULTS
point	Proposition 2.1
change-of-coordinates	Proposition 2.2, Proposition 2.3, Proposition 2.4, Proposition 2.5
tangent vector	Proposition 3.1, Proposition 3.2, Corollary 3.3
metric	Proposition 3.4, Proposition 3.5
normal vector	Proposition 3.6, Corollary 3.7
curve	Proposition 4.2
geodesic	Theorem 4.3, Proposition 4.5
geodesic distance	Corollary 4.6
exponential map	Corollary 4.4
logarithmic map	Corollary 4.7
parallel transport	Proposition 4.8
gradient	Proposition 5.1, Corollary 5.3
Hessian	Proposition 5.2
retraction and vector transport	Proposition 6.4, Proposition 6.5, Proposition 6.6
steepest descent	Algorithm 1, Algorithm 2
Newton method	Algorithm 3
conjugate gradient	Algorithm 4
quasi-Newton	Algorithm 5

TABLE 1. Guide to results.

and algorithms in Table 1 are considerably simpler compared to those in [2, 24, 34]. We will not need to solve quadratic eigenvalue problems, nor compute $\exp/\cos/\sin/\text{sinc}$ of nonnormal matrices, nor even EVD or SVD except in cases when they can be trivially obtained. Aside from standard matrix arithmetic, our optimization algorithms require just two operations:

- (iii) all differential geometric objects and operations can be computed with at most a QR decomposition and an exponentiation of a skew-symmetric matrix,

$$\exp\left(\begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right), \quad B \in \mathbb{R}^{k \times (n-k)}, \quad (5)$$

which may in turn be computed in time $O(nk(n-k))$ with a specialized algorithm based on Strang splitting.

The problem of computing matrix exponential has been thoroughly studied and there is a plethora of algorithms [37, 52], certainly more so than other transcendental matrix functions like cosine, sine, or sinc [37]. For normal matrices, matrix exponentiation is a well-conditioned problem — the numerical issues described in [52] only occur with nonnormal matrices. For us, $\begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}$ is skew-symmetric and thus normal; in fact its exponential will always be an orthogonal matrix.

There are other algorithmic advantages afforded by (4) that are difficult to explain without context and will be discussed alongside the algorithms in Section 7 and numerical results in Section 8. In particular, our algorithms will work with what we call “effective gradients,” “effective Newton steps,” “effective conjugate directions,” etc — these are all matrices of size $k \times (n-k)$ like the matrix B in (5), i.e., they have the intrinsic dimension of $\text{Gr}(k, n)$. With this we would also like to add a note of caution. One cannot infer an accurate estimate of computational complexity based on a simple dimension count of the models in Table 2. There are many differential geometric objects and operations involved in an algorithm, such as those in Table 1, and not just points. The matrices arising in actual computations are highly structured and the computational cost depends heavily on the specific problem.

2. POINTS AND CHANGE-OF-COORDINATES

We begin by exhibiting a diffeomorphism to justify the involution model, showing that as *smooth manifolds*, $\text{Gr}(k, n)$ and $\{Q \in \text{O}(n) : Q^\top = Q, \text{tr}(Q) = 2k - n\}$ are the same. In the next section, we will show that if we equip the latter with appropriate Riemannian metrics, then as *Riemannian manifolds*, they are also the same, i.e., the diffeomorphism is an isometry. The practically minded may simply take this as establishing a system of matrix coordinates for points on $\text{Gr}(k, n)$.

Proposition 2.1 (Points). *Let $k = 1, \dots, n$. Then the map*

$$\begin{aligned} \varphi : \text{Gr}(k, n) &\rightarrow \{Q \in \text{O}(n) : Q^\top = Q, \text{tr}(Q) = 2k - n\}, \\ \varphi(\mathbb{W}) &= P_{\mathbb{W}} - P_{\mathbb{W}^\perp}, \end{aligned} \tag{7}$$

is a diffeomorphism with $\varphi^{-1}(Q) = \text{im}(Y_Q)$ where $Y_Q \in \text{V}(k, n)$ is as in (6). Here $P_{\mathbb{W}}$ denotes the orthogonal projection matrix to the subspace \mathbb{W} .

Proof. One can check that $Q = P_{\mathbb{W}} - P_{\mathbb{W}^\perp}$ is symmetric, orthogonal, and has trace $2k - n$. So the map φ is well-defined. If we write $\psi(Q) = \text{im}(Y_Q)$, then $\varphi(\psi(Q)) = Q$ and $\psi(\varphi(\mathbb{W})) = \mathbb{W}$, so $\psi = \varphi^{-1}$. To see that φ is smooth, we may choose any local coordinates, say, represent $\mathbb{W} \in \text{Gr}(k, n)$ in terms of any orthonormal basis $W = [w_1, \dots, w_k] \in \text{V}(k, n)$ and observe that

$$\varphi(\mathbb{W}) = 2WW^\top - I,$$

which is smooth. With a linear change-of-coordinates, we may assume that

$$W = \begin{bmatrix} I_k \\ 0 \end{bmatrix}.$$

The differential $(d\varphi)_{\mathbb{W}}$ is given by the (clearly invertible) linear map

$$(d\varphi)_{\mathbb{W}} \left(\begin{bmatrix} 0 \\ X \end{bmatrix} \right) = 2 \left(\begin{bmatrix} I_k \\ 0 \end{bmatrix} \begin{bmatrix} 0 & X^\top \end{bmatrix} + \begin{bmatrix} 0 \\ X \end{bmatrix} \begin{bmatrix} I_k & 0 \end{bmatrix} \right) = 2 \begin{bmatrix} 0 & X^\top \\ X & 0 \end{bmatrix}$$

for all $X \in \mathbb{R}^{(n-k) \times k}$. So φ is a diffeomorphism. \square

Since the manifolds in Table 2 are all diffeomorphic to $\text{Gr}(k, n)$, they are diffeomorphic to each other. Our next results are not intended to establish that they are diffeomorphic but to construct these diffeomorphisms and their inverses explicitly, so that we may switch to and from the other systems of coordinates easily.

In the next proposition, $\llbracket V \rrbracket = \left\{ V \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} : Q_1 \in \text{O}(k), Q_2 \in \text{O}(n-k) \right\}$ denotes equivalence class in $\text{O}(n)/(\text{O}(k) \times \text{O}(n-k))$.

Proposition 2.2 (Change-of-coordinates I). *Let $k = 1, \dots, n$. Then*

$$\begin{aligned} \varphi_1 : \text{O}(n)/(\text{O}(k) \times \text{O}(n-k)) &\rightarrow \{Q \in \text{O}(n) : Q^\top = Q, \text{tr}(Q) = 2k - n\}, \\ \varphi_1(\llbracket V \rrbracket) &= V^\top I_{k, n-k} V \end{aligned}$$

is a diffeomorphism with $\varphi_1^{-1}(Q) = \llbracket V_Q \rrbracket$ with $V_Q \in \text{O}(n)$ as in (6).

Proof. Note that $Q = V_1 I_{k, n-k} V_1^\top = V_2 I_{k, n-k} V_2^\top$ iff

$$V_2 = V_1 \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$$

for some $(Q_1, Q_2) \in \text{O}(k) \times \text{O}(n-k)$ iff $\llbracket V_1 \rrbracket = \llbracket V_2 \rrbracket$. Hence both φ_1 and φ_1^{-1} are well-defined and are inverses of each other. Observe that φ_1 is induced from the map

$$\tilde{\varphi}_1 : \text{O}(n) \rightarrow \{Q \in \text{O}(n) : Q^\top = Q, \text{tr}(Q) = 2k - n\}, \quad \tilde{\varphi}_1(V) = V^\top I_{k, n-k} V,$$

which is a surjective submersion. The proof that φ_1^{-1} is well-defined shows that the fibers of $\tilde{\varphi}_1$ are exactly the $O(k) \times O(n-k)$ -orbits in $O(n)$. Hence φ_1 , as the composition of $\tilde{\varphi}_1$ and the quotient map $O(n) \rightarrow O(n)/(O(k) \times O(n-k))$, is a diffeomorphism. \square

The next result explains the resemblance between the projection and involution models — each is a scaled and translated copy of the other. The scaling and translation are judiciously chosen so that orthogonal projections become symmetric involutions, and this seemingly innocuous difference will have a significant impact on the numerical stability of Grassmannian optimization algorithms.

Proposition 2.3 (Change-of-coordinates II). *Let $k = 1, \dots, n$. Then*

$$\begin{aligned} \varphi_2 : \{P \in \mathbb{R}^{n \times n} : P^\top = P = P^2, \operatorname{tr}(P) = k\} &\rightarrow \{Q \in O(n) : Q^\top = Q, \operatorname{tr}(Q) = 2k - n\}, \\ \varphi_2(P) &= 2P - I \end{aligned}$$

is a diffeomorphism with $\varphi_2^{-1}(Q) = \frac{1}{2}(I + Q)$.

Proof. Note that $2P - I = P - P^\perp$ where P^\perp is the projection onto the orthogonal complement of $\operatorname{im}(P)$, so both φ_2 and φ_2^{-1} are well-defined. They are clearly diffeomorphisms and are inverses to each other. \square

In the next proposition, $\llbracket Y \rrbracket = \{YQ : Q \in O(k)\}$ denotes equivalence class in $V(k, n)/O(k)$.

Proposition 2.4 (Change-of-coordinates III). *Let $k = 1, \dots, n$. Then*

$$\begin{aligned} \varphi_3 : V(k, n)/O(k) &\rightarrow \{Q \in O(n) : Q^\top = Q, \operatorname{tr}(Q) = 2k - n\}, \\ \varphi_3(\llbracket Y \rrbracket) &= 2YY^\top - I \end{aligned}$$

is a diffeomorphism with $\varphi_3^{-1}(Q) = \llbracket Y_Q \rrbracket$ with $Y_Q \in V(k, n)$ as in (6).

Proof. Given $\llbracket Y \rrbracket \in V(k, n)/O(k)$, the matrix YY^\top is the projection matrix onto the k -dimensional subspace $\operatorname{im}(Y) \in \operatorname{Gr}(k, n)$. Hence φ_3 is a well-defined map by Proposition 2.3. To show that its inverse is given by $\psi_3(Q) = \llbracket Y_Q \rrbracket$, observe that any $Y \in V(k, n)$ can be extended to a full orthogonal matrix $V := [Y, Y^\perp] \in O(n)$ and we have

$$V^\top Y = \begin{bmatrix} I_k \\ 0 \end{bmatrix}, \quad Q = 2YY^\top - I = V \begin{bmatrix} 2I_k & 0 \\ 0 & 0 \end{bmatrix} V^\top - I = VI_{k, n-k}V^\top.$$

This implies that $\psi_3 \circ \varphi_3(\llbracket Y \rrbracket) = \llbracket Y_Q \rrbracket = \llbracket Y \rrbracket$. That φ_3 is a diffeomorphism follows from the same argument in the proof of Proposition 2.1. \square

In the next proposition, $\llbracket A \rrbracket = \{AX : X \in \operatorname{GL}(k)\}$ denotes equivalence class in $\mathbb{R}_k^{n \times k}/\operatorname{GL}(k)$. Also, we write $A = Y_A R_A$ for the QR factorization of $A \in \mathbb{R}_k^{n \times k}$, i.e., $Y_A \in V(k, n)$ and $R_A \in \mathbb{R}^{k \times k}$ is upper triangular.

Proposition 2.5 (Change-of-coordinates IV). *Let $k = 1, \dots, n$. Then*

$$\begin{aligned} \varphi_4 : \mathbb{R}_k^{n \times k}/\operatorname{GL}(k) &\rightarrow \{Q \in O(n) : Q^\top = Q, \operatorname{tr}(Q) = 2k - n\}, \\ \varphi_4(\llbracket A \rrbracket) &= 2Y_A Y_A^\top - I \end{aligned}$$

is a diffeomorphism with $\varphi_4^{-1}(Q) = \llbracket Y_Q \rrbracket$ with Y_Q as in (6).

Proof. First observe that $V(k, n) \subseteq \mathbb{R}_k^{n \times k}$ and the inclusion map $V(k, n) \hookrightarrow \mathbb{R}_k^{n \times k}$ induces a diffeomorphism $V(k, n)/O(k) \cong \mathbb{R}_k^{n \times k}/\operatorname{GL}(k)$ — if we identify them, then φ_4^{-1} becomes φ_3^{-1} in Proposition 2.4 and is thus a diffeomorphism. It follows that φ_4 is a diffeomorphism. That the maps are inverses to each other follows from the same argument in the proof of Proposition 2.4. \square

The maps $\varphi, \varphi_1, \varphi_2, \varphi_3, \varphi_4$ allow one to transform an optimization problem formulated in terms of abstract k -dimensional subspaces or in terms of one of the first four models in Table 2 into a mathematically (but not computationally) equivalent problem in terms of the involution model. Note that these are change-of-coordinate maps for *points* — they are good for translating expressions that involve only points on $\text{Gr}(k, n)$. In particular, one cannot simply apply these maps to the analytic expressions for other differential geometric objects and operations in [2, 24, 34] and obtain corresponding expressions for the involution model. Deriving these requires considerable effort and would take up the next three sections.

Henceforth we will identify the Grassmannian with the involution model:

$$\text{Gr}(k, n) := \{Q \in \text{O}(n) : Q^\top = Q, \text{tr}(Q) = 2k - n\},$$

i.e., in the rest of our article, points on $\text{Gr}(k, n)$ are symmetric orthogonal matrices of trace $2k - n$. With this, the well-known isomorphism

$$\text{Gr}(k, n) \cong \text{Gr}(n - k, n), \quad (8)$$

which we will need later, is simply given by the map $Q \mapsto -Q$.

3. METRIC, TANGENTS, AND NORMALS

The simple observation in Lemma 1.1 implies that a neighborhood of any point $Q \in \text{Gr}(k, n)$ is just like a neighborhood of the special point $I_{k, n-k} = \text{diag}(I_k, -I_{n-k}) \in \text{Gr}(k, n)$. Consequently, objects like tangent spaces and curves at Q can be determined by simply determining them at $I_{k, n-k}$. Although $\text{Gr}(k, n)$ is not a Lie group, the involution model, which models it as a linear section of $\text{O}(n)$, allows certain characteristics of a Lie group to be retained. Here $I_{k, n-k}$ has a role similar to that of the identity element in a Lie group.

We will provide three different expressions for vectors in the *tangent space* $\mathbb{T}_Q \text{Gr}(k, n)$ at a point $Q \in \text{Gr}(k, n)$: an implicit form (9) as traceless symmetric matrices that anticommutes with Q and two explicit forms (10), (11) parameterized by $k \times (n - k)$ matrices. Recall from Lemma 1.1 that any $Q \in \text{Gr}(k, n)$ has an eigendecomposition of the form $Q = VI_{k, n-k}V^\top$ for some $V \in \text{O}(n)$.

Proposition 3.1 (Tangent space I). *Let $Q \in \text{Gr}(k, n)$ with eigendecomposition $Q = VI_{k, n-k}V^\top$. The tangent space of $\text{Gr}(k, n)$ at Q is given by*

$$\mathbb{T}_Q \text{Gr}(k, n) = \{X \in \mathbb{R}^{n \times n} : X^\top = X, XQ + QX = 0, \text{tr}(X) = 0\} \quad (9)$$

$$= \left\{ V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top \in \mathbb{R}^{n \times n} : B \in \mathbb{R}^{k \times (n-k)} \right\} \quad (10)$$

$$= \left\{ QV \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} V^\top \in \mathbb{R}^{n \times n} : B \in \mathbb{R}^{k \times (n-k)} \right\}. \quad (11)$$

Proof. By definition, a curve γ in $\text{Gr}(k, n)$ passing through Q satisfies

$$\gamma(t)^\top - \gamma(t) = 0, \quad \gamma(t)^\top \gamma(t) = I_n, \quad \text{tr}(\gamma(t)) = 2k - n, \quad t \in (-\varepsilon, \varepsilon),$$

together with the initial condition $\gamma(0) = Q$. Differentiating these equations at $t = 0$, we get

$$\dot{\gamma}(0)^\top - \dot{\gamma}(0) = 0, \quad \dot{\gamma}(0)^\top Q + Q^\top \dot{\gamma}(0) = 0, \quad \text{tr}(\dot{\gamma}(0)) = 0,$$

from which (9) follows. Now take $X \in \mathbb{T}_Q \text{Gr}(k, n)$. By (9), $V^\top X V I_{k, n-k} = V^\top (XQ) V$ is skew-symmetric and $V^\top X V$ is symmetric. Partition

$$V^\top X V = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}, \quad A \in \mathbb{R}^{k \times k}, \quad B \in \mathbb{R}^{k \times (n-k)}, \quad C \in \mathbb{R}^{(n-k) \times (n-k)}.$$

Note that A and C are symmetric matrices since X is. So if

$$V^\top X V I_{k, n-k} = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} = \begin{bmatrix} A & -B \\ B^\top & -C \end{bmatrix}$$

is skew-symmetric, then we must have $A = 0$ and $C = 0$ and we obtain (10). Since $Q = VI_{k,n-k}V^\top$ and $Q = Q^\top$, (11) follows from (10) by writing $V = QVI_{k,n-k}$. \square

The implicit form in (9) is inconvenient in algorithms. Of the two explicit forms (10) and (11), the description in (10) is evidently more economical, involving only V , as opposed to both Q and V as in (11). Henceforth, (10) will be our preferred choice and we will assume that a tangent vector at $Q \in \text{Gr}(k, n)$ always takes the form

$$X = V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top, \quad (12)$$

for some $B \in \mathbb{R}^{k \times (n-k)}$. This description appears to depend on the eigenbasis V , which is not unique, as Q has many repeated eigenvalues. The next proposition, which relates two representations of the same tangent vector with respect to two different V 's, guarantees that the tangent space obtained will nonetheless be the same regardless of the choice of V .

Proposition 3.2 (Tangent vectors). *If $V_1 I_{k,n-k} V_1^\top = Q = V_2 I_{k,n-k} V_2^\top$, then any $X \in \mathbb{T}_Q \text{Gr}(k, n)$ can be written as*

$$X = V_2 \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V_2^\top = V_1 \begin{bmatrix} 0 & Q_1 B Q_2^\top \\ Q_2 B^\top Q_1^\top & 0 \end{bmatrix} V_1^\top,$$

for some $Q_1 \in O(k)$ and $Q_2 \in O(n-k)$ such that

$$V_2 = V_1 \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}. \quad (13)$$

Proof. This is a consequence of the fact that $V_1 I_{k,n-k} V_1^\top = Q = V_2 I_{k,n-k} V_2^\top$ iff there exist $Q_1 \in O(k)$ and $Q_2 \in O(n-k)$ such that (13) holds. \square

Another consequence of using (10) is that the tangent space at any point Q is a copy of the tangent space at $I_{k,n-k}$, conjugated by any eigenbasis V of Q ; by Proposition 3.2, this is independent of the choice of V .

Corollary 3.3 (Tangent space II). *The tangent space at $I_{k,n-k}$ is*

$$\mathbb{T}_{I_{k,n-k}} \text{Gr}(k, n) = \left\{ \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} : B \in \mathbb{R}^{k \times (n-k)} \right\}.$$

For any $Q \in \text{Gr}(k, n)$ with eigendecomposition $Q = VI_{k,n-k}V^\top$,

$$\mathbb{T}_Q \text{Gr}(k, n) = V(\mathbb{T}_{I_{k,n-k}} \text{Gr}(k, n))V^\top.$$

With the tangent spaces characterized, we may now define an inner product $\langle \cdot, \cdot \rangle_Q$ on each $\mathbb{T}_Q \text{Gr}(k, n)$ that varies smoothly over all $Q \in \text{Gr}(k, n)$, i.e., a *Riemannian metric*. With the involution model, $\text{Gr}(k, n)$ is a submanifold of $O(n)$ and there is a natural choice, namely, the Riemannian metric inherited from that on $O(n)$.

Proposition 3.4 (Riemannian metric). *Let $Q \in \text{Gr}(k, n)$ with $Q = VI_{k,n-k}V^\top$ and*

$$X = V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top, \quad Y = V \begin{bmatrix} 0 & C \\ C^\top & 0 \end{bmatrix} V^\top \in \mathbb{T}_Q \text{Gr}(k, n).$$

Then

$$\langle X, Y \rangle_Q := \text{tr}(XY) = 2 \text{tr}(B^\top C) \quad (14)$$

defines a Riemannian metric. The corresponding Riemannian norm is

$$\|X\|_Q := \sqrt{\langle X, X \rangle_Q} = \|X\|_F = \sqrt{2} \|B\|_F. \quad (15)$$

The Riemannian metric in (14) is induced by the unique (up to a positive constant multiple) bi-invariant Riemannian metric on $O(n)$:

$$g_Q(X, Y) := \text{tr}(X^\top Y), \quad Q \in O(n), \quad X, Y \in \mathbb{T}_Q O(n).$$

Here bi-invariance may be taken to mean

$$g_{V_1 Q V_2^\top}(V_1 X V_2^\top, V_1 Y V_2^\top) = g_Q(X, Y)$$

for all $Q, V_1, V_2 \in O(n)$ and $X, Y \in \mathbb{T}_Q O(n)$.

There are also natural Riemannian metrics [2, 24, 34] on the other four models in Table 2 but they differ from each other by a constant. As such, it is not possible for us to choose our metric (14) so that the diffeomorphisms in Propositions 2.2–2.5 are all isometry but we do have the next best thing.

Proposition 3.5 (Isometry). *All models in Table 2 are, up to a constant factor, isometric as Riemannian manifolds.*

Proof. We verify that the diffeomorphism φ_1 in Proposition 2.2 gives an isometry between the orthogonal model and the involution model up a constant factor of 8. A tangent vector [24, Equation 2.30] at a point $\llbracket V \rrbracket \in O(n)/(O(k) \times O(n-k))$ takes the form

$$V \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} \in \mathbb{T}_{\llbracket V \rrbracket} O(n)/(O(k) \times O(n-k)), \quad B \in \mathbb{R}^{k \times (n-k)};$$

and the Riemannian metric [24, Equation 2.31] on $O(n)/(O(k) \times O(n-k))$ is given by

$$g_{\llbracket V \rrbracket} \left(V \begin{bmatrix} 0 & B_1 \\ -B_1^\top & 0 \end{bmatrix}, V \begin{bmatrix} 0 & B_2 \\ -B_2^\top & 0 \end{bmatrix} \right) = \text{tr}(B_1^\top B_2).$$

At I_n , the differential can be computed by

$$(d\varphi_1)_{\llbracket I_n \rrbracket} \left(I_n \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} \right) = 2I_{k, n-k} \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} = 2 \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix}.$$

Since both g and $\langle \cdot, \cdot \rangle$ are invariant under left multiplication by $O(n)$, we have

$$\left\langle (d\varphi_1)_{\llbracket V \rrbracket} \left(V \begin{bmatrix} 0 & B_1 \\ -B_1^\top & 0 \end{bmatrix} \right), (d\varphi_1)_{\llbracket V \rrbracket} \left(V \begin{bmatrix} 0 & B_2 \\ -B_2^\top & 0 \end{bmatrix} \right) \right\rangle_{\varphi_1(\llbracket V \rrbracket)} = 8 \text{tr}(B_1^\top B_2).$$

The proofs for $\varphi_2, \varphi_3, \varphi_4$ are similar and thus omitted. \square

As the above proof shows, the diffeomorphism φ_1 may be easily made an isometry of the orthogonal and involution models by simply changing our metric in (14) to “ $\langle X, Y \rangle_Q := \frac{1}{8} \text{tr}(XY)$.” Had we wanted to make φ_2 into an isometry of the projection and involution models, we would have to choose “ $\langle X, Y \rangle_Q := \frac{1}{2} \text{tr}(XY)$ ” instead. We see no reason to favor any single existing model and we stick to our choice of metric in (14).

In the involution model, $\text{Gr}(k, n) \subseteq O(n)$ as a smoothly embedded submanifold and every point $Q \in \text{Gr}(k, n)$ has a *normal space* $\mathbb{N}_Q \text{Gr}(k, n)$. We will next determine the expressions for normal vectors.

Proposition 3.6 (Normal space). *Let $Q \in \text{Gr}(k, n)$ with $Q = VI_{k, n-k}V^\top$. The normal space of $\text{Gr}(k, n)$ at Q is given by*

$$\mathbb{N}_Q \text{Gr}(k, n) = \left\{ V \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} V^\top \in \mathbb{R}^{n \times n} : \begin{array}{ll} \Lambda_1 \in \mathbb{R}^{k \times k}, & \Lambda_2 \in \mathbb{R}^{(n-k) \times (n-k)} \\ \Lambda_1^\top = -\Lambda_1, & \Lambda_2^\top = -\Lambda_2 \end{array} \right\}.$$

Proof. The tangent space of a point $Q \in O(n)$ is given by

$$\mathbb{T}_Q O(n) = \{Q\Lambda \in \mathbb{R}^{n \times n} : \Lambda^\top = -\Lambda\}.$$

A tangent vector $Q\Lambda \in \mathbb{T}_Q O(n)$ is normal to $\text{Gr}(k, n)$ at Q iff

$$0 = \langle X, Q\Lambda \rangle_Q = \text{tr}(X^\top Q\Lambda),$$

for all $X \in \mathbb{T}_Q \text{Gr}(k, n)$. By (12), $X = V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top$ where $Q = VI_{k, n-k} V^\top$. Thus

$$\text{tr} \left(V^\top \Lambda V \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} \right) = 0 \quad (16)$$

for all $B \in \mathbb{R}^{k \times (n-k)}$. Since (16) must hold for all $B \in \mathbb{R}^{k \times (n-k)}$, we must have

$$\Lambda = V \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} V^\top, \quad (17)$$

for some skew-symmetric matrices $\Lambda_1 \in \mathbb{R}^{k \times k}$, $\Lambda_2 \in \mathbb{R}^{(n-k) \times (n-k)}$, and therefore,

$$Q\Lambda = VI_{k, n-k} V^\top \Lambda = V \begin{bmatrix} \Lambda_1 & 0 \\ 0 & -\Lambda_2 \end{bmatrix} V^\top.$$

Conversely, any Λ of the form in (17) must satisfy (16). \square

Propositions 3.1 and 3.6 allow us to explicitly decompose the tangent space of $O(n)$ at a point $Q \in \text{Gr}(k, n)$ into

$$\begin{aligned} \mathbb{T}_Q O(n) &= \mathbb{T}_Q \text{Gr}(k, n) \oplus \mathbb{N}_Q \text{Gr}(k, n), \\ Q\Lambda &= QV \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} V^\top + V \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} V^\top. \end{aligned}$$

For later purposes, it will be useful to give explicit expressions for the two projection maps.

Corollary 3.7 (Projection maps). *Let $Q \in \text{Gr}(k, n)$ with $Q = VI_{k, n-k} V^\top$ and*

$$\text{proj}_Q^{\mathbb{T}} : \mathbb{T}_Q O(n) \rightarrow \mathbb{T}_Q \text{Gr}(k, n), \quad \text{proj}_Q^{\mathbb{N}} : \mathbb{T}_Q O(n) \rightarrow \mathbb{N}_Q \text{Gr}(k, n)$$

be the projection maps onto the tangent and normal spaces of $\text{Gr}(k, n)$ respectively. Then

$$\begin{aligned} \text{proj}_Q^{\mathbb{T}}(Q\Lambda) &= \frac{1}{2}(Q\Lambda - \Lambda Q) = \frac{1}{2}V(S + S^\top)V^\top, \\ \text{proj}_Q^{\mathbb{N}}(Q\Lambda) &= \frac{1}{2}(Q\Lambda + \Lambda Q) = \frac{1}{2}V(S - S^\top)V^\top, \end{aligned} \quad (18)$$

for any decomposition $Q\Lambda = VSV^\top$ where $S \in \mathbb{R}^{n \times n}$ is such that $I_{k, n-k}S$ is skew-symmetric.

Proof. We see from Propositions 3.1 and 3.6 that the maps are well defined, i.e., $\frac{1}{2}(Q\Lambda - \Lambda Q) \in \mathbb{T}_Q \text{Gr}(k, n)$ and $\frac{1}{2}(Q\Lambda + \Lambda Q) \in \mathbb{N}_Q \text{Gr}(k, n)$, and the images are orthogonal as

$$\langle Q\Lambda - \Lambda Q, Q\Lambda + \Lambda Q \rangle_Q = 0.$$

The alternative expressions follow from taking $S = I_{k, n-k} V^\top \Lambda V$. \square

4. EXPONENTIAL MAP, GEODESIC, AND PARALLEL TRANSPORT

An explicit and easily computable formula for a geodesic curve is indispensable in most Riemannian optimization algorithms. By Lemma 1.1, any $Q \in \text{Gr}(k, n)$ can be eigendecomposed as $VI_{k, n-k}V^\top$ for some $V \in \text{O}(n)$. So a curve γ in $\text{Gr}(k, n)$ takes the form

$$\gamma(t) = V(t)I_{k, n-k}V(t)^\top, \quad (19)$$

with $V(t)$ a curve in $\text{O}(n)$ that can in turn be written as

$$V(t) = V \exp(\Lambda(t)), \quad (20)$$

where $\Lambda(t)$ is a curve in the space of $n \times n$ skew-symmetric matrices, $\Lambda(0) = 0$, and $V(0) = V$. We will show in Proposition 4.2 that in the involution model the curve $\Lambda(t)$ takes a particularly simple form. We first prove a useful lemma using the CS decomposition [29, 59].

Lemma 4.1. *Let $\Lambda \in \mathbb{R}^{n \times n}$ be skew-symmetric. Then there exist $B \in \mathbb{R}^{k \times (n-k)}$ and two skew-symmetric matrices $\Lambda_1 \in \mathbb{R}^{k \times k}$, $\Lambda_2 \in \mathbb{R}^{(n-k) \times (n-k)}$ such that*

$$\exp(\Lambda) = \exp\left(\begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right) \exp\left(\begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix}\right). \quad (21)$$

Proof. By (8), we may assume $k \leq n/2$. Let the CS decomposition of $Q := \exp(\Lambda) \in \text{O}(n)$ be

$$Q = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-2k} \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & Z \end{bmatrix}^\top, \quad (22)$$

where $U, W \in \text{O}(k)$, $V, Z \in \text{O}(n-k)$, $\Theta = \text{diag}(\theta_1, \dots, \theta_k)$ with $\theta_i \in [0, 2\pi]$, $i = 1, \dots, k$. Next we will show that we may always choose U, W, V, Z to have determinant one, i.e., $U, W \in \text{SO}(k)$, $V, Z \in \text{SO}(n-k)$. To see this, note that

$$\det(Q) = \det(\exp(\Lambda)) = 1, \quad \det \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-2k} \end{bmatrix} = 1,$$

and so we must have $\det(U) \det(W) \det(V) \det(Z) = 1$. If $\det(U) = \det(W) = \det(V) = \det(Z) = 1$, then we are done. If either two of them, or all four of them have determinant -1 , let $\hat{I}_j := \text{diag}(1, \dots, -1, \dots, 1)$, i.e., the identity matrix with its j th diagonal entry replaced by -1 , then inserting \hat{I}_1 and \hat{I}_{k+1} in (22) allows us to change the signs of the determinants at will. For example, if $\det(U) = \det(Z) = -1$, $\det(W) = \det(V) = 1$, then

$$Q = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \hat{I}_1 \hat{I}_{k+1} \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-2k} \end{bmatrix} \hat{I}_{k+1} \hat{I}_1 \begin{bmatrix} W & 0 \\ 0 & Z \end{bmatrix}^\top = \begin{bmatrix} U' & 0 \\ 0 & V' \end{bmatrix} \begin{bmatrix} \cos \Theta' & \sin \Theta' & 0 \\ -\sin \Theta' & \cos \Theta' & 0 \\ 0 & 0 & I_{n-2k} \end{bmatrix} \begin{bmatrix} W' & 0 \\ 0 & Z' \end{bmatrix}^\top$$

and the new matrices U', V', W', Z' now satisfy $\det(U') = \det(V') = \det(W') = \det(Z') = 1$. It is easy to check that in all cases, we may assume that $U, W \in \text{SO}(k)$, $V, Z \in \text{SO}(n-k)$ without loss of generality. Now

$$\begin{aligned} \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-2k} \end{bmatrix} &= \exp\left(\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} 0 & \Theta & 0 \\ -\Theta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}^\top\right) \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \\ &= \exp\left(\begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right) \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}, \end{aligned}$$

where $B := U[\Theta, 0]V^\top \in \mathbb{R}^{k \times (n-k)}$ with $0 \in \mathbb{R}^{k \times (n-2k)}$. As $UW^\top \in \text{SO}(k)$ and $VZ^\top \in \text{SO}(n-k)$, we can find skew symmetric matrices Λ_1, Λ_2 with $\exp(\Lambda_1) = UW^\top$ and $\exp(\Lambda_2) = VZ^\top$ as required. \square

Proposition 4.2 (Curve). *Let $Q \in \text{Gr}(k, n)$ with eigendecomposition $Q = VI_{k, n-k}V^\top$. Then a curve $\gamma(t)$ in $\text{Gr}(k, n)$ through Q may be expressed as*

$$\gamma(t) = V \exp \left(\begin{bmatrix} 0 & B(t) \\ -B(t)^\top & 0 \end{bmatrix} \right) I_{k, n-k} \exp \left(\begin{bmatrix} 0 & -B(t) \\ B(t)^\top & 0 \end{bmatrix} \right) V^\top \quad (23)$$

for some curve $B(t)$ in $\mathbb{R}^{k \times (n-k)}$ through the zero matrix.

Proof. By (19) and (20), we have

$$\gamma(t) = V \exp(\Lambda(t)) I_{k, n-k} \exp(-\Lambda(t)) V^\top.$$

By Lemma 4.1, we may write

$$\exp(\Lambda(t)) = \exp \left(\begin{bmatrix} 0 & B(t) \\ -B(t)^\top & 0 \end{bmatrix} \right) \exp \left(\begin{bmatrix} \Lambda_1(t) & 0 \\ 0 & \Lambda_2(t) \end{bmatrix} \right),$$

which gives the desired parametrization in (23). \square

Proposition 4.2 yields another way to obtain the expression for tangent vectors in (12). Differentiating the curve in (23) at $t = 0$, we get

$$\dot{\gamma}(0) = V \left(\begin{bmatrix} 0 & -2\dot{B}(0) \\ -2\dot{B}(0)^\top & 0 \end{bmatrix} \right) V^\top \in \mathbb{T}_Q \text{Gr}(k, n).$$

Choosing $B(t)$ to be any curve in $\mathbb{R}^{k \times (n-k)}$ with $B(0) = 0$ and $\dot{B}(0) = -B/2$, we obtain (12).

The key ingredient in most manifold optimization algorithms is the geodesic at a point in a direction. In [24], the discussion regarding geodesics on the Grassmannian is brief: Essentially, it says that because a geodesic on the Stiefel manifold $V(k, n)$ takes the form $Q \exp(t\Lambda)$, a geodesic on the Grassmannian $V(k, n)/O(k)$ takes the form $\llbracket Q \exp(t\Lambda) \rrbracket$. It is hard to be more specific when one uses the Stiefel model. On the other hand, when we use the involution model, the expression (25) in the next theorem describes a geodesic precisely, and any point on γ can be evaluated with a single QR decomposition (to obtain V , see Section 7.1) and a single matrix exponentiation (the two exponents are transposes of each other).

Theorem 4.3 (Geodesics I). *Let $Q \in \text{Gr}(k, n)$ and $X \in \mathbb{T}_Q \text{Gr}(k, n)$ with*

$$Q = VI_{k, n-k}V^\top, \quad X = V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top. \quad (24)$$

The geodesic γ emanating from Q in the direction X is given by

$$\gamma(t) = V \exp \left(\frac{t}{2} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} \right) I_{k, n-k} \exp \left(\frac{t}{2} \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} \right) V^\top. \quad (25)$$

The differential equation for γ is

$$\gamma(t)^\top \ddot{\gamma}(t) - \ddot{\gamma}(t)^\top \gamma(t) = 0, \quad \gamma(0) = Q, \quad \dot{\gamma}(0) = X. \quad (26)$$

Proof. By Proposition 4.2, any curve through Q must take the form

$$\gamma(t) = V \exp \left(\begin{bmatrix} 0 & B(t) \\ -B(t)^\top & 0 \end{bmatrix} \right) I_{k, n-k} \exp \left(\begin{bmatrix} 0 & -B(t) \\ B(t)^\top & 0 \end{bmatrix} \right) V^\top,$$

where $B(0) = 0$. Since γ is in the direction X , we have that $\dot{\gamma}(0) = X$, and thus $\dot{B}(0) = -B/2$. It remains to employ the fact that as a geodesic, γ is a critical curve of the length functional

$$L(\gamma) := \int_0^1 \|\dot{\gamma}(t)\|_{\gamma(t)} dt$$

where the Riemannian norm is as in (15). Let $\varepsilon > 0$. Consider a variation of $\gamma(t)$ with respect to a C^1 -curve $C(t)$ in $\mathbb{R}^{k \times (n-k)}$:

$$\gamma_\varepsilon(t) = V \exp \left(\begin{bmatrix} 0 & B(t) + \varepsilon C(t) \\ -B(t)^\top - \varepsilon C(t)^\top & 0 \end{bmatrix} \right) I_{k,n-k} \exp \left(\begin{bmatrix} 0 & -B(t) - \varepsilon C(t) \\ B(t)^\top + \varepsilon C(t)^\top & 0 \end{bmatrix} \right) V^\top.$$

We require $C(0) = C(1) = 0$ so that γ_ε is a variation of γ with fixed end points. The tangent vector of γ_ε at time t is given by

$$V \exp \left(\begin{bmatrix} 0 & B(t) + \varepsilon C(t) \\ -B(t)^\top - \varepsilon C(t)^\top & 0 \end{bmatrix} \right) \left(-2 \begin{bmatrix} 0 & \dot{B}(t) + \varepsilon \dot{C}(t) \\ \dot{B}(t) + \varepsilon \dot{C}(t)^\top & 0 \end{bmatrix} \right) \exp \left(\begin{bmatrix} 0 & -B(t) - \varepsilon C(t) \\ B(t)^\top + \varepsilon C(t)^\top & 0 \end{bmatrix} \right) V^\top$$

and so $\|\dot{\gamma}_\varepsilon(t)\|_{\gamma(t)} = 2\sqrt{2}\|\dot{B}(t) + \varepsilon\dot{C}(t)\|_F$ where $\|\cdot\|_F$ denotes Frobenius norm. Hence,

$$0 = \frac{d}{d\varepsilon} L(\gamma_\varepsilon(t)) \Big|_{\varepsilon=0} = 2\sqrt{2} \int_0^1 \frac{\text{tr}(\dot{B}(t)^\top \dot{C}(t))}{\|\dot{B}(t)\|_F} dt.$$

As $\gamma(t)$ is a geodesic, $\|\dot{\gamma}(t)\|_{\gamma(t)}$ and thus $\|\dot{B}(t)\|_F$ must be a constant $K > 0$. Therefore, we have

$$0 = \frac{1}{K} \int_0^1 \text{tr}(\dot{B}(t)^\top \dot{C}(t)) dt = -\frac{1}{K} \int_0^1 \text{tr}(\ddot{B}(t)^\top C(t)) dt,$$

implying that $\ddot{B}(t) = 0$ and thus $B(t) = t\dot{B}(0) = -tB/2$. Lastly, since

$$\begin{aligned} \dot{\gamma}(t) &= V \exp \left(\begin{bmatrix} 0 & B(t) \\ -B(t)^\top & 0 \end{bmatrix} \right) \left(-2 \begin{bmatrix} 0 & \dot{B}(t) \\ \dot{B}(t)^\top & 0 \end{bmatrix} \right) \exp \left(\begin{bmatrix} 0 & -B(t) \\ B(t)^\top & 0 \end{bmatrix} \right) V^\top, \\ \ddot{\gamma}(t) &= V \exp \left(\begin{bmatrix} 0 & B(t) \\ -B(t)^\top & 0 \end{bmatrix} \right) \left(-4 \begin{bmatrix} \dot{B}(t)\dot{B}(t)^\top & \\ & -\dot{B}(t)^\top \dot{B}(t) \end{bmatrix} - 2 \begin{bmatrix} 0 & \ddot{B}(t) \\ \ddot{B}(t)^\top & 0 \end{bmatrix} \right) \exp \left(\begin{bmatrix} 0 & -B(t) \\ B(t)^\top & 0 \end{bmatrix} \right) V^\top, \end{aligned} \quad (27)$$

and the differential equation for a geodesic curve γ is

$$\text{proj}_{\gamma(t)}^\top(\ddot{\gamma}) = 0, \quad \gamma(0) = Q, \quad \dot{\gamma}(0) = X,$$

we obtain (26) from the expression for tangent projection in (18). \square

Theorem 4.3 also gives the *exponential map* of X .

Corollary 4.4 (Exponential map). *Let $Q \in \text{Gr}(k, n)$ and $X \in \mathbb{T}_Q \text{Gr}(k, n)$ be as in (24). Then*

$$\text{exp}_Q(X) := \gamma(1) = V \exp \left(\frac{1}{2} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} \right) I_{k,n-k} \exp \left(\frac{1}{2} \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} \right) V^\top. \quad (28)$$

The length of the geodesic segment from $\gamma(0) = 0$ to $\gamma(1) = \text{exp}_Q(X)$ is

$$L(\gamma) = \|X\|_F = \sqrt{2}\|B\|_F. \quad (29)$$

The Grassmannian is geodesically complete and so any two points can be joined by a length-minimizing geodesic. In the next proposition, we will derive an explicit expression for such a geodesic in the involution model. By (8), there will be no loss of generality in assuming that $k \leq n/2$ in the following — if $k > n/2$, then we just replace k by $n - k$.

Proposition 4.5 (Geodesics II). *Let $k \leq n/2$. Let $Q_0, Q_1 \in \text{Gr}(k, n)$ with eigendecompositions $Q_0 = V_0 I_{k,n-k} V_0^\top$ and $Q_1 = V_1 I_{k,n-k} V_1^\top$. Let the CS decomposition of $V_0^\top V_1 \in O(n)$ be*

$$V_0^\top V_1 = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-2k} \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & Z \end{bmatrix}^\top \quad (30)$$

where $U, W \in O(k)$, $V, Z \in O(n - k)$, $\Theta = \text{diag}(\theta_1, \dots, \theta_k) \in \mathbb{R}^{k \times k}$. Then the geodesic γ connecting Q_0 to Q_1 is

$$\gamma(t) = V_0 \exp \left(\frac{t}{2} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} \right) I_{k,n-k} \exp \left(\frac{t}{2} \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} \right) V_0^\top,$$

where $B = -2U[\Theta, 0]V^\top \in \mathbb{R}^{k \times (n-k)}$ with $\Theta \in \mathbb{R}^{k \times (n-2k)}$.

Proof. By Theorem 4.3, γ is a geodesic curve emanating from $\gamma(0) = V_0 I_{k,n-k} V_0^\top = Q_0$. It remains to verify that

$$\gamma(1) = V_0 \exp\left(\frac{1}{2} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix}\right) I_{k,n-k} \exp\left(\frac{1}{2} \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right) V_0^\top = Q_1,$$

when $B = -2U[\Theta, 0]V^\top$. Substituting the expression for B ,

$$\begin{aligned} \gamma(1) &= V_0 \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \exp\left(\begin{bmatrix} 0 & \Theta & 0 \\ -\Theta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) I_{k,n-k} \exp\left(\begin{bmatrix} 0 & -\Theta & 0 \\ \Theta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) \begin{bmatrix} U^\top & 0 \\ 0 & V^\top \end{bmatrix} V_0^\top \\ &= V_0 \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-2k} \end{bmatrix} I_{k,n-k} \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} U^\top & 0 \\ 0 & V^\top \end{bmatrix} V_0^\top \\ &= V_0 \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-2k} \end{bmatrix} \begin{bmatrix} W^\top & 0 \\ 0 & Z^\top \end{bmatrix} I_{k,n-k} \begin{bmatrix} W & 0 \\ 0 & Z \end{bmatrix} \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} U^\top & 0 \\ 0 & V^\top \end{bmatrix} V_0^\top \end{aligned}$$

where the last equality holds because we have

$$I_{k,n-k} = \begin{bmatrix} W^\top & 0 \\ 0 & Z^\top \end{bmatrix} I_{k,n-k} \begin{bmatrix} W & 0 \\ 0 & Z \end{bmatrix}$$

whenever $W \in O(k)$ and $Z \in O(n-k)$. By (30), the last expression of $\gamma(1)$ equals

$$V_0(V_0^\top V_1) I_{k,n-k} (V_0^\top V_1)^\top V_0^\top = V_1 I_{k,n-k} V_1^\top = Q_1. \quad \square$$

The geodesic expression in Proposition 4.5 requires a CS decomposition [29, 59] and is more expensive to evaluate than the one in Theorem 4.3. Nevertheless, we do not need Proposition 4.5 for our optimization algorithms in Section 7, although its next corollary could be useful if one wants to design proximal gradient methods in the involution model.

Corollary 4.6 (Geodesic distance). *The geodesic distance between $Q_0, Q_1 \in \text{Gr}(k, n)$ is given by*

$$d(Q_0, Q_1) = 2\sqrt{2} \left(\sum_{i=1}^k \sigma_i(B)^2 \right)^{1/2} = 2\sqrt{2} \left(\sum_{i=1}^k \theta_i \right)^{1/2} \quad (31)$$

where $B \in \mathbb{R}^{k \times (n-k)}$ and $\Theta \in \mathbb{R}^{k \times k}$ are as in Proposition 4.5.

Proof. By (29), $L(\gamma) = \sqrt{2}\|B\|_F = 2\sqrt{2}\|\Theta\|_F$ with $B = -2U[\Theta, 0]V^\top$ as in Proposition 4.5. \square

The last expression in (31) differs from the expression in [24, Section 4.3] by a factor of $2\sqrt{2}$, which is exactly what we expect since the metrics in the involution and orthogonal models differ by a factor of $(2\sqrt{2})^2 = 8$, as we saw in the proof of Proposition 3.5.

The notion of a logarithmic map is somewhat less standard and we remind readers of its definition. Given a Riemannian manifold M and a point $x \in M$, there exists some $r > 0$ such that the exponential map $\exp_x : B_r(0) \rightarrow M$ is a diffeomorphism on the ball $B_r(0) \subseteq \mathbb{T}_x M$ of radius r centered at the origin [22, Theorem 3.7]. The *logarithm map*, sometimes called the *inverse exponential map*, is then defined on the diffeomorphic image $\exp_x(B_r(0)) \subseteq M$ by

$$\log_x : \exp_x(B_r(0)) \rightarrow \mathbb{T}_x M, \quad \log_x(v) := \exp_x^{-1}(v)$$

for all $v \in \exp_x(B_r(0))$. The largest r so that \exp_x is a diffeomorphism on $B_r(0)$ is the *injectivity radius* at x and its infimum over all $x \in M$ is the injectivity radius of M .

Corollary 4.7 (Logarithmic map). *Let $Q_0, Q_1 \in \text{Gr}(k, n)$ be such that $d(Q_0, Q_1) < \sqrt{2}\pi$. Let $V_0, V_1 \in \text{O}(n)$, and $B \in \mathbb{R}^{k \times (n-k)}$ be as in Proposition 4.5. The logarithmic map at Q_0 of Q_1 is*

$$\log_{Q_0}(Q_1) = V_0 \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} V_0^\top.$$

Proof. The injectivity radius of $\text{Gr}(k, n)$ is well known to be $\pi/2$ [67]. Write $B_r(0) = \{X \in \mathbb{T}_{Q_0} \text{Gr}(k, n) : \|X\|_Q < r\}$ and $B_r^d(Q_0) = \{Q \in \text{Gr}(k, n) : d(Q_0, Q) < r\}$. By Corollaries 4.4 and 4.6,

$$\exp_{Q_0}(B_{\pi/2}(0)) = B_{\sqrt{2}\pi}^d(Q_0).$$

By Corollary 4.4 and Proposition 4.5, $\log_{Q_0} : B_{\sqrt{2}\pi}(Q_0) \rightarrow \text{Gr}(k, n)$ has the required expression. \square

We end this section with the expression for the parallel transport of a vector Y along a geodesic γ at a point Q in the direction X . This will be an essential ingredient for conjugate gradient and Newton methods in the involution model (see Algorithms 3 and 4).

Proposition 4.8 (Parallel transport). *Let $Q \in \text{Gr}(k, n)$ and $X, Y \in \mathbb{T}_Q \text{Gr}(k, n)$ with*

$$Q = VI_{k, n-k}V^\top, \quad X = V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top, \quad Y = V \begin{bmatrix} 0 & C \\ C^\top & 0 \end{bmatrix} V^\top,$$

where $V \in \text{O}(n)$ and $B, C \in \mathbb{R}^{k \times (n-k)}$. Let γ be a geodesic curve emanating from Q in the direction X . Then the parallel transport of Y along γ is

$$Y(t) = V \exp\left(\frac{t}{2} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix}\right) \begin{bmatrix} 0 & C \\ C^\top & 0 \end{bmatrix} \exp\left(\frac{t}{2} \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right) V^\top. \quad (32)$$

Proof. Let γ be parametrized as in (25). A vector field $Y(t)$ that is parallel along $\gamma(t)$ may, by (12), be written in the form

$$Y(t) = V \exp\left(\frac{t}{2} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix}\right) \begin{bmatrix} 0 & C(t) \\ C(t)^\top & 0 \end{bmatrix} \exp\left(\frac{t}{2} \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right) V^\top$$

for some curve $C(t)$ in $\mathbb{R}^{k \times (n-k)}$ with $C(0) = C$. Differentiating $Y(t)$ gives

$$\dot{Y}(t) = V \exp\left(\frac{t}{2} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix}\right) \begin{bmatrix} -\frac{1}{2}(BC(t)^\top + C(t)B^\top) & \dot{C}(t) \\ \dot{C}(t)^\top & \frac{1}{2}(B^\top C(t) + C(t)^\top B) \end{bmatrix} \exp\left(\frac{t}{2} \begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right) V^\top.$$

Since $Y(t)$ is parallel along $\gamma(t)$, we must have

$$\text{proj}_{\gamma(t)}^\mathbb{T}(\dot{Y}(t)) = 0,$$

which implies that $\dot{C}(t) = 0$ and thus $C(t) = C(0) = C$, giving us (32). \square

A word about our notation for parallel transport, or rather, the lack of one. Note that $Y(t)$ depends on γ and to indicate this dependence, we may write $Y_\gamma(t)$. Other common notations include $\tau_t Y$ [33], $P_t^\gamma Y$ [40], $\gamma_s^t(Y)$ [46] ($s = 0$ for us) but there is no single standard notation.

5. GRADIENT AND HESSIAN

We now derive expressions for the Riemannian gradient and Hessian of a C^2 function $f : \text{Gr}(k, n) \rightarrow \mathbb{R}$ in the involution model with (10) for tangent vectors. As a reminder, this means:

$$\begin{aligned} \text{Gr}(k, n) &= \{Q \in \mathbb{R}^{n \times n} : Q^\top Q = I, Q^\top = Q, \text{tr}(Q) = 2k - n\}, \\ \mathbb{T}_Q \text{Gr}(k, n) &= \left\{ V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top \in \mathbb{R}^{n \times n} : B \in \mathbb{R}^{k \times (n-k)} \right\}, \end{aligned} \quad (33)$$

where $Q = VI_{k, n-k}V^\top$.

Let $Q \in \text{Gr}(k, n)$. Then the *Riemannian gradient* ∇f at Q is a tangent vector $\nabla f(Q) \in \mathbb{T}_Q \text{Gr}(k, n)$ and, depending on context, the *Riemannian Hessian* at Q is a bilinear map:

$$\nabla^2 f(Q) : \mathbb{T}_Q \text{Gr}(k, n) \times \mathbb{T}_Q \text{Gr}(k, n) \rightarrow \mathbb{R}.$$

Proposition 5.1 (Riemannian gradient I). *Let $f : \text{Gr}(k, n) \rightarrow \mathbb{R}$ be C^1 . For any $Q \in \text{Gr}(k, n)$, write*

$$f_Q := \left[\frac{\partial f}{\partial q_{ij}}(Q) \right]_{i,j=1}^n \in \mathbb{R}^{n \times n}. \quad (34)$$

Then

$$\nabla f(Q) = \frac{1}{4} [f_Q + f_Q^\top - Q(f_Q + f_Q^\top)Q]. \quad (35)$$

Proof. The projection of $QX \in \mathbb{T}_Q \mathbb{R}^{n \times n}$ to $\mathbb{T}_Q \text{O}(n)$ is $Q(X - X^\top)/2$. Therefore the projection of $f_Q \in \mathbb{T}_Q \mathbb{R}^{n \times n}$ to $\mathbb{T}_Q \text{O}(n)$ is $(f_Q - Qf_Q^\top Q)/2$. Composing this with the projection of $\mathbb{T}_Q \text{O}(n)$ to $\mathbb{T}_Q \text{Gr}(k, n)$ given in (18), we get

$$\nabla f(Q) = \text{proj}_Q^\mathbb{T} \left(\frac{f_Q - Qf_Q^\top Q}{2} \right) = \frac{1}{4} (f_Q + f_Q^\top - Qf_Q Q - Qf_Q^\top Q)$$

as required. \square

Proposition 5.2 (Riemannian Hessian I). *Let $f : \text{Gr}(k, n) \rightarrow \mathbb{R}$ be C^2 . For any $Q = VI_{k,n-k}V^\top \in \text{Gr}(k, n)$, let f_Q be as in (34) and*

$$f_{QQ}(X) := \left[\sum_{i,j=1}^n \left(\frac{\partial^2 f}{\partial q_{ij} \partial q_{kl}}(Q) \right) x_{ij} \right]_{k,l=1}^n, \quad f_{QQ}(X, Y) := \sum_{i,j,k,l=1}^n \left(\frac{\partial^2 f}{\partial q_{ij} \partial q_{kl}}(Q) \right) x_{ij} y_{kl}.$$

As a bilinear map, the Hessian of f at Q is given by

$$\nabla^2 f(Q)(X, Y) = f_{QQ}(X, Y) - \frac{1}{2} \text{tr}(f_Q^\top Q(XY + YX)) \quad (36)$$

for any $X, Y \in \mathbb{T}_Q \text{Gr}(k, n)$.

Proof. Let γ be a geodesic curve emanating from Q in the direction $X \in \mathbb{T}_Q \text{Gr}(k, n)$. Then

$$\nabla^2 f(Q)(X, X) = \left. \frac{d^2}{dt^2} f(\gamma(t)) \right|_{t=0} = \left. \frac{d}{dt} \text{tr}(f_{\gamma(t)}^\top \dot{\gamma}(t)) \right|_{t=0} = f_{QQ}(X) + \text{tr}(f_Q^\top \ddot{\gamma}(0)).$$

Since $\gamma(t)$ is given by (25),

$$\ddot{\gamma}(0) = V \begin{bmatrix} -BB^\top & 0 \\ 0 & B^\top B \end{bmatrix} V^\top = -Q\dot{\gamma}(0)^2$$

and so

$$\nabla^2 f(Q)(X, X) = f_{QQ}(X) - \text{tr}(f_Q^\top QX^2).$$

To obtain $\nabla^2 f(Q)$ as a bilinear map, we simply polarize the quadratic form above:

$$\begin{aligned} \nabla^2 f(Q)(X, Y) &= \frac{1}{2} [\nabla^2 f(Q)(X + Y, X + Y) - \nabla^2 f(Q)(X, X) - \nabla^2 f(Q)(Y, Y)] \\ &= \frac{1}{2} \left[f_{QQ}(X + Y) - f_{QQ}(X) - f_{QQ}(Y) - \text{tr}(f_Q^\top Q(XY + YX)) \right] \\ &= f_{QQ}(X, Y) - \frac{1}{2} \text{tr}(f_Q^\top Q(XY + YX)). \end{aligned}$$

\square

Our optimization algorithms require that we parameterize our tangent space as in (33) and we need to express $\nabla f(Q)$ in such a form. This can be easily accomplished. Let $E_{ij} \in \mathbb{R}^{k \times (n-k)}$ be the matrix whose (i, j) entry is zero and other entries are one. Let

$$X_{ij} := V \begin{bmatrix} 0 & E_{ij} \\ E_{ij}^\top & 0 \end{bmatrix} V^\top \in \mathbb{T}_Q \text{Gr}(k, n). \quad (37)$$

Then $\mathcal{B}_Q := \{X_{ij} : i = 1, \dots, k, j = 1, \dots, n-k\}$ is an orthogonal (but not orthonormal since Riemannian norm $\|X_{ij}\|_Q = 1/\sqrt{2}$) basis of $\mathbb{T}_Q \text{Gr}(k, n)$.

Corollary 5.3 (Riemannian gradient II). *Let f, Q, f_Q be as in Propositions 5.1. If we partition*

$$V^\top (f_Q + f_Q^\top) V = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}, \quad (38)$$

where $A \in \mathbb{R}^{k \times k}$, $B \in \mathbb{R}^{k \times (n-k)}$, $C \in \mathbb{R}^{(n-k) \times (n-k)}$, then

$$\nabla f(Q) = \frac{1}{2} V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top. \quad (39)$$

Proof. By (38), we may rewrite (35) as

$$\nabla f(Q) = \frac{1}{4} \left(V \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} V^\top - V \begin{bmatrix} A & -B \\ -B^\top & C \end{bmatrix} V^\top \right) = \frac{1}{2} V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top. \quad \square$$

In our optimization algorithms, (38) is how we actually compute Riemannian gradients. Note that in the basis \mathcal{B}_Q , the gradient of f is essentially given by the matrix $B/2 \in \mathbb{R}^{k \times (n-k)}$. So in algorithms that rely only on Riemannian gradients, we just need the top right block B , but the other blocks A and C would appear implicitly in the Riemannian Hessians.

We may order the basis \mathcal{B}_Q lexicographically (note that X_{ij} 's are indexed by two indices), then the bilinear form $\nabla^2 f(Q)$ has the matrix representation

$$H_Q := \begin{bmatrix} \nabla^2 f(Q)(X_{11}, X_{11}) & \nabla^2 f(Q)(X_{11}, X_{12}) & \dots & \nabla^2 f(Q)(X_{11}, X_{k, n-k}) \\ \nabla^2 f(Q)(X_{12}, X_{11}) & \nabla^2 f(Q)(X_{12}, X_{12}) & \dots & \nabla^2 f(Q)(X_{12}, X_{k, n-k}) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^2 f(Q)(X_{k, n-k}, X_{11}) & \nabla^2 f(Q)(X_{k, n-k}, X_{12}) & \dots & \nabla^2 f(Q)(X_{k, n-k}, X_{k, n-k}) \end{bmatrix}. \quad (40)$$

In practice, the evaluation of H_Q may be simplified; we will discuss this in Section 7.3. To summarize, in the lexicographically ordered basis \mathcal{B}_Q ,

$$[\nabla f(Q)]_{\mathcal{B}_Q} = \frac{1}{2} \text{vec}(B) \in \mathbb{R}^{k(n-k)}, \quad [\nabla^2 f(Q)]_{\mathcal{B}_Q} = H_Q \in \mathbb{R}^{k(n-k) \times k(n-k)},$$

and the Newton step $S \in \mathbb{R}^{k \times (n-k)}$ is given by the linear system

$$H_Q \text{vec}(S) = -\frac{1}{2} \text{vec}(B). \quad (41)$$

6. RETRACTION MAP AND VECTOR TRANSPORT

Up till this point, everything that we have discussed is authentic Riemannian geometry, even though we have used extrinsic coordinates to obtain expressions in terms of matrices and matrix operations. This section is a departure, we will discuss two notions created for sole use in manifold optimization: *retraction maps* [5, 57] and *vector transports* [3]. They are relaxations of exponential maps and parallel transports respectively and are intended to be pragmatic substitutes in situations where these Riemannian operations are either too difficult to compute (e.g., requiring the exponential of a nonnormal matrix) or unavailable in closed form (e.g., parallel transport on a Stiefel

manifold). While the involution model does not suffer from either of these problems, retraction algorithms could still serve as a good option for initializing Riemannian optimization algorithms.

As these definitions are not found in the Riemannian geometry literature, we state a version of [3, Definitions 4.1.1 and 8.1.1] below for easy reference.

Definition 6.1. A map $R : \mathbb{T}M \rightarrow M$, $(x, v) \mapsto R_x(v)$ is a *retraction* map if it satisfies the following two conditions:

- (a) $R_x(0) = x$ for all $x \in M$;
- (b) $dR_x(0) : \mathbb{T}_x M \rightarrow \mathbb{T}_x M$ is the identity map for all $x \in M$.

A map $T : \mathbb{T}M \oplus \mathbb{T}M \rightarrow \mathbb{T}M$ associated to a retraction map R is a *vector transport* if it satisfies the following three conditions:

- (i) $T(x, v, w) = (R_x(v), T_{x,v}(w))$ for all $x \in M$ and $v, w \in \mathbb{T}_x M$;
- (ii) $T_{x,0}(w) = w$ for all $x \in M$ and $w \in \mathbb{T}_x M$;
- (iii) $T_{x,v}(a_1 w_1 + a_2 w_2) = a_1 T_{x,v}(w_1) + a_2 T_{x,v}(w_2)$ for all $a_1, a_2 \in \mathbb{R}$, $x \in M$, and $v, w_1, w_2 \in \mathbb{T}_x M$.

Here $\mathbb{T}M \oplus \mathbb{T}M$ is a direct sum of vector bundles and each element is parametrized by a point $x \in M$ and two tangent vectors $v, w \in \mathbb{T}_x M$. The condition (i) says that the vector transport T is compatible with its retraction map R , and also defines the map $T_{x,v} : \mathbb{T}_x M \rightarrow \mathbb{T}_x M$. Note that v is the direction to move in while w is the vector to be transported.

For the purpose of optimization, we just need R and T to be well-defined on a neighbourhood of $M \cong \{(x, 0) \in \mathbb{T}M\} \subseteq \mathbb{T}M$ and $M \cong \{(x, 0, 0) \in \mathbb{T}M \oplus \mathbb{T}M\} \subseteq \mathbb{T}M \oplus \mathbb{T}M$ respectively. If R and T are C^1 maps, then various optimization algorithms relying on R and T can be shown to converge [3], possibly under the additional assumption that M has nonnegative [18] or bounded sectional curvature [62]. In particular, these results apply in our case since being a compact symmetric space, $\text{Gr}(k, n)$ has both nonnegative and bounded sectional curvature [14, 71].

Example 6.2 (Projection as retraction). For a manifold M embedded in Euclidean space \mathbb{R}^n or $\mathbb{R}^{m \times n}$, we may regard tangent vectors in $\mathbb{T}_x M$ to be of the form $x + v$. In this case an example of a retraction map is given by the projection of tangent vectors onto M ,

$$R_x(v) = \underset{y \in M}{\operatorname{argmin}} \|x + v - y\|,$$

where $\|\cdot\|$ is either the 2- or Frobenius norm. By [4, Lemma 3.1], the map R_x is well-defined for small v and is a retraction.

We will give three retraction maps for $\text{Gr}(k, n)$ that are readily computable in the involution model with EVD, block QR, and Cayley transform respectively. The latter two are inspired by similar maps defined for the projection model in [34] although our motivations are somewhat different.

We begin by showing how one may compute the projection $\operatorname{argmin}\{\|A - Q\|_F : Q \in \text{Gr}(k, n)\}$ for an arbitrary matrix $A \in \mathbb{R}^{n \times n}$ in the involution model, a result that may be of independent interest.

Lemma 6.3. *Let $A \in \mathbb{R}^{n \times n}$ and*

$$\frac{A + A^\top}{2} = V D V^\top \tag{42}$$

be an eigendecomposition with $V \in \text{O}(n)$ and $D = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$, $\lambda_1 \geq \dots \geq \lambda_n$. Then $Q = V I_{k, n-k} V^\top$ is a minimizer of

$$\min\{\|A - Q\|_F : Q^\top Q = I, Q^\top = Q, \operatorname{tr}(Q) = 2k - n\}.$$

Proof. Since Q is symmetric, $\|A - Q\|_F^2 = \|(A + A^\top)/2 - Q\|_F^2 + \|(A - A^\top)/2\|_F^2$, a best approximation to A is also a best approximation to $(A + A^\top)/2$. By (42), $\|(A + A^\top)/2 - Q\|_F = \|D - V^\top Q V\|_F$ and so for a best approximation $V^\top Q V$ must be a diagonal matrix. Since the eigenvalues $\delta_1, \dots, \delta_n$ of

a symmetric orthogonal Q must be ± 1 and $\text{tr}(Q) = 2k - n$, the multiplicities of $+1$ and -1 are k and $n - k$ respectively. By assumption, $\lambda_1 \geq \dots \geq \lambda_n$, so

$$\min_{\delta_1 + \dots + \delta_n = 2k - n} (\lambda_1 - \delta_1)^2 + \dots + (\lambda_n - \delta_n)^2$$

is attained when $\delta_1 = \dots = \delta_k = +1$ and $\delta_{k+1} = \dots = \delta_n = -1$. Hence $V^\top Q V = \text{diag}(\delta_1, \dots, \delta_n) = I_{k, n-k}$ as required. \square

It is clear from the proof, which is a variation of standard arguments [37, Section 8.1], that a minimizer is not unique if and only if $\lambda_k = \lambda_{k+1}$, i.e., the k th and $(k+1)$ th eigenvalues of $(A + A^\top)/2$ coincide. Since any $Q \in \text{Gr}(k, n)$ by definition has $\lambda_k = +1 \neq -1 = \lambda_{k+1}$, the projection is always unique in a small enough neighborhood of Q in $\mathbb{R}^{n \times n}$.

In the following, let $\mathcal{E} : \mathbb{R}^{n \times n} \rightarrow O(n)$ be the map that takes any $A \in \mathbb{R}^{n \times n}$ to an orthogonal matrix of eigenvectors of $(A + A^\top)/2$.

Proposition 6.4 (Retraction I). *Let $Q \in \text{Gr}(k, n)$ and $X, Y \in \mathbb{T}_Q \text{Gr}(k, n)$ with*

$$Q = V I_{k, n-k} V^\top, \quad X = V \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V^\top, \quad Y = V \begin{bmatrix} 0 & C \\ C^\top & 0 \end{bmatrix} V^\top, \quad (43)$$

where $V \in O(n)$ and $B, C \in \mathbb{R}^{k \times (n-k)}$. Then

$$R_Q^\mathcal{E}(X) = V \mathcal{E} \left(\begin{bmatrix} I & B \\ B^\top & -I \end{bmatrix} \right) I_{k, n-k} \mathcal{E} \left(\begin{bmatrix} I & B \\ B^\top & -I \end{bmatrix} \right)^\top V^\top$$

defines a retraction and

$$T_Q^\mathcal{E}(X, Y) = V \mathcal{E} \left(\begin{bmatrix} I & B \\ B^\top & -I \end{bmatrix} \right) \begin{bmatrix} 0 & C \\ C^\top & 0 \end{bmatrix} \mathcal{E} \left(\begin{bmatrix} I & B \\ B^\top & -I \end{bmatrix} \right)^\top V^\top$$

defines a vector transport.

Proof. It follows from Lemma 6.3 that $R_Q^\mathcal{E}$ defines a projection. The properties in Definition 6.1 are routine to verify. \square

As we will see later, the exponential map in our Riemannian algorithms may be computed in $O(nk(n-k))$ time, so a retraction map that requires an EVD offers no advantage. Furthermore, the eigenvector map \mathcal{E} is generally discontinuous [45], which can present a problem. One alternative would be to approximate the map \mathcal{E} with a QR decomposition — one should think of this as the first step of Francis's QR algorithm for EVD. In fact, we will not even require a full QR decomposition, a 2×2 block QR decomposition suffices. Let $\mathcal{Q} : \mathbb{R}^{n \times n} \rightarrow O(n)$ be a map that takes a matrix A to its orthogonal factor in a 2×2 block QR decomposition, i.e.,

$$A = \mathcal{Q}(A) \begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix}, \quad R_1 \in \mathbb{R}^{k \times k}, \quad R_2 \in \mathbb{R}^{k \times (n-k)}, \quad R_3 \in \mathbb{R}^{(n-k) \times (n-k)}.$$

Note that $\mathcal{Q}(A)$ is an orthogonal matrix but the second factor just needs to be block upper triangular, i.e., R_1 and R_3 are not required to be upper triangular matrices. We could compute $\mathcal{Q}(A)$ with, say, the first k steps of Householder QR applied to A .

Proposition 6.5 (Retraction II). *Let $Q \in \text{Gr}(k, n)$ and $X, Y \in \mathbb{T}_Q \text{Gr}(k, n)$ be as in (43). If \mathcal{Q} is well-defined and differentiable near $I_{k, n-k}$ and $\mathcal{Q}(I_{k, n-k}) = I$, then*

$$R_Q^\mathcal{Q}(X) = V \mathcal{Q} \left(\frac{1}{2} \begin{bmatrix} I & B \\ B^\top & -I \end{bmatrix} \right) I_{k, n-k} \mathcal{Q} \left(\frac{1}{2} \begin{bmatrix} I & B \\ B^\top & -I \end{bmatrix} \right)^\top V^\top$$

defines a retraction and

$$T_Q^\mathcal{Q}(X, Y) = V \mathcal{Q} \left(\frac{1}{2} \begin{bmatrix} I & B \\ B^\top & -I \end{bmatrix} \right) \begin{bmatrix} 0 & C \\ C^\top & 0 \end{bmatrix} \mathcal{Q} \left(\frac{1}{2} \begin{bmatrix} I & B \\ B^\top & -I \end{bmatrix} \right)^\top V^\top$$

defines a vector transport.

Proof. Only property (b) in Definition 6.1 is not immediate and requires checking. Let the following be a block QR decomposition:

$$\frac{1}{2} \begin{bmatrix} I & tB \\ tB^\top & -I \end{bmatrix} = \begin{bmatrix} Q_1(t) & Q_2(t) \\ Q_3(t) & Q_4(t) \end{bmatrix} \begin{bmatrix} R_1(t) & R_2(t) \\ 0 & R_3(t) \end{bmatrix} = Q(t)R(t), \quad (44)$$

with $Q(t) \in O(n)$. Since $Q(t)Q(t)^\top = 1$ and $Q(0) = I$, $Q'(0)$ is skew-symmetric and

$$\left. \frac{d}{dt} Q(t) I_{k,n-k} Q(t)^\top \right|_{t=0} = \begin{bmatrix} Q'_1(0) + Q'_1(0)^\top & -Q'_2(0) + Q'_3(0)^\top \\ Q'_3(0) - Q'_2(0)^\top & -Q'_4(0) - Q'_4(0)^\top \end{bmatrix} = \begin{bmatrix} 0 & 2Q'_3(0)^\top \\ 2Q'_3(0) & 0 \end{bmatrix}.$$

Comparing the (1, 1) and (2, 1) entries in (44), we get

$$Q_1(t)R_1(t) = I, \quad Q_3(t)R_1(t) = tB^\top/2.$$

Hence $Q_3(t) = tB^\top Q_1(t)/2$, $Q'_3(0) = B^\top Q_1(0)/2 = B^\top/2$, and we get

$$\left. \frac{d}{dt} Q(t) I_{k,n-k} Q(t)^\top \right|_{t=0} = \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix},$$

as required. \square

If we use a first-order Padé approximation $\exp(X) \approx (I+X)(I-X)^{-1}$ for the matrix exponential terms in the exponential map (28) and parallel transport (32), we obtain another retraction map and vector transport. This Padé approximation is the well-known *Cayley transform* \mathcal{C} , which takes a skew-symmetric matrix to an orthogonal matrix and vice versa:

$$\mathcal{C} : \Lambda^2(\mathbb{R}^n) \rightarrow O(n), \quad \Lambda \rightarrow (I + \Lambda)(I - \Lambda)^{-1}.$$

Proposition 6.6 (Retraction III). *Let $Q \in \text{Gr}(k, n)$ and $X, Y \in \mathbb{T}_Q \text{Gr}(k, n)$ be as in (43). Then*

$$R_Q^{\mathcal{C}}(X) = V \mathcal{C} \left(\frac{1}{4} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} \right) I_{k,n-k} \mathcal{C} \left(\frac{1}{4} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} \right)^\top V^\top$$

defines a retraction and

$$T_Q^{\mathcal{C}}(X, Y) = V \mathcal{C} \left(\frac{1}{4} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} \right) \begin{bmatrix} 0 & C \\ C^\top & 0 \end{bmatrix} \mathcal{C} \left(\frac{1}{4} \begin{bmatrix} 0 & -B \\ B^\top & 0 \end{bmatrix} \right)^\top V^\top$$

defines a vector transport.

Proof. Again, only property (b) in Definition 6.1 is not immediate and requires checking. But this is routine we omit the details. \square

7. ALGORITHMS

We will now discuss optimization algorithms for minimizing a function $f : \text{Gr}(k, n) \rightarrow \mathbb{R}$ in the involution model. In principle, this is equivalent to a quadratically constrained optimization problem in n^2 variables $[q_{ij}]_{i,j=1}^n = Q \in \mathbb{R}^{n \times n}$:

$$\begin{aligned} & \text{minimize} && f(Q) \\ & \text{subject to} && Q^\top Q = I, Q^\top = Q, \text{tr}(Q) = 2k - n. \end{aligned} \quad (45)$$

Nevertheless, if one attempts to minimize any of the objective functions f in Section 8 by treating (45) as a general nonlinear constrained optimization problem using, say, the MATLAB Optimization Toolbox, every available method — interior point, trust region, sequential quadratic programming, active set — will fail without even finding a feasible point, never mind a minimizer. The Riemannian geometric objects and operations of the last few sections are essential to solving (45).

We will distinguish between two types of optimization algorithms. The *retraction algorithms*, as its name implies, will be based on various retractions and vector transports discussed in Section 6.

The *Riemannian algorithms*, on the other hand, are built upon true Riemannian geodesics and parallel transports discussed in Section 4. Both types of algorithms will rely on the materials on points in Section 2, tangent vectors and metric in Section 3, and Riemannian gradients and Hessians in Section 5.

For both types of algorithms, the involution model offers one significant advantage over other existing models. By (39) and (28), at a point $Q \in \text{Gr}(k, n)$ and in a direction $X \in \mathbb{T}_Q \text{Gr}(k, n)$, the Riemannian gradient and the exponential map are

$$\nabla f(Q) = V \begin{bmatrix} 0 & G/2 \\ G^\top/2 & 0 \end{bmatrix} V^\top, \quad \exp_Q(X) = V \exp \left(\begin{bmatrix} 0 & -B/2 \\ B^\top/2 & 0 \end{bmatrix} \right) I_{k, n-k} \exp \left(\begin{bmatrix} 0 & B/2 \\ -B^\top/2 & 0 \end{bmatrix} \right) V^\top$$

respectively. In the involution model, explicit parallel transport and exponential map can be avoided. Instead of $\nabla f(Q)$ and $\exp_Q(X)$, it suffices to work with the matrices $G, B \in \mathbb{R}^{k \times (n-k)}$ that we will call *effective gradient* and *effective step* respectively, and doing so leads to extraordinarily simple and straightforward expressions in our algorithms. We will highlight this simplicity at appropriate junctures in Sections 7.2 and 7.3. Aside from simplicity, a more important consequence is that all key computations in our algorithms are performed at the *intrinsic dimension* of $\text{Gr}(k, n)$. Our steepest descent direction, conjugate direction, Barzilai–Borwein step, Newton step, quasi-Newton step, etc, would all be represented as $k(n-k)$ -dimensional objects. This is a feature not found in the algorithms of [2, 24, 34].

7.1. Initialization, eigendecomposition, and exponentiation. We begin by addressing three issues that we will frequently encounter in our optimization algorithms.

First observe that it is trivial to generate a point $Q \in \text{Gr}(k, n)$ in the involution model: Take any orthogonal matrix $V \in O(n)$, generated by say a QR decomposition of a random $n \times n$ matrix. Then we always have $Q := VI_{k, n-k}V^\top \in \text{Gr}(k, n)$. We may easily generate as many random feasible initial points for our algorithms as we desire or simply take $I_{k, n-k}$ as our initial point.

The inverse operation of obtaining a $V \in O(n)$ from a given $Q \in \text{Gr}(k, n)$ so that $Q = VI_{k, n-k}V^\top$ seems more expensive as it appears to require an EVD. In fact, by the following observation, the cost is the same — a single QR decomposition.

Lemma 7.1. *Let $Q \in \mathbb{R}^{n \times n}$ with $Q^\top Q = I$, $Q^\top = Q$, $\text{tr}(Q) = 2k - n$. If*

$$\frac{1}{2}(I + Q) = V \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}, \quad V \in O(n), \quad R_1 \in \mathbb{R}^{k \times k}, \quad R_2 \in \mathbb{R}^{k \times (n-k)}, \quad (46)$$

is a QR decomposition, then $Q = VI_{k, n-k}V^\top$.

Proof. Recall from (6) that for such a Q , we may write $V = [Y, Z]$ where $Y \in V(k, n)$ and $Z \in V(n-k, n)$ are a +1-eigenbasis and a -1-eigenbasis of Q respectively. By Proposition 2.3, $\frac{1}{2}(I + Q)$ is the projection matrix onto the +1-eigenspace $\text{im}(Y) = \text{im}(\frac{1}{2}(I + Q))$, i.e., Y is an orthonormal column basis for $\frac{1}{2}(I + Q)$ and is therefore given by its condensed QR decomposition. As for Z , note that any orthonormal basis for $\text{im}(Y)^\perp$ would serve the role, i.e., Z can be obtained from the full QR decomposition. In summary,

$$\frac{1}{2}(I + Q) = Y \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = [Y \quad Z] \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}.$$

As a sanity check, note that

$$\frac{1}{2}(I + Q) = YY^\top = [Y \quad Z] \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Z \end{bmatrix} = V \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} V^\top,$$

and therefore

$$Q = V \begin{bmatrix} I_k & 0 \\ 0 & -I_{n-k} \end{bmatrix} V^\top = VI_{k, n-k}V^\top.$$

□

Our expressions for tangent vector, exponential map, geodesic, parallel transport, retraction, etc, at a point $Q \in \text{Gr}(k, n)$ all involve its matrix of eigenvectors $V \in \text{O}(n)$. So Lemma 7.1 plays an important role in our algorithms. In practice, numerical stability considerations in the presence of rounding errors [21, Section 3.5.2] require that we perform our QR decomposition with *column pivoting* so that (46) becomes

$$\frac{1}{2}(I + Q) = V \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix} \Pi^\top$$

where Π is a permutation matrix. This does not affect our proof above; in particular, note that we have no need for R_1 nor R_2 nor Π in any of our algorithms.

The most expensive step in our Riemannian algorithms is the evaluation

$$B \mapsto \exp\left(\begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right) \quad (47)$$

for $B \in \mathbb{R}^{k \times (n-k)}$. General algorithms for computing matrix exponential [37, 52] do not exploit structures aside from normality. There are specialized algorithms that take advantage of skew-symmetry³ [12] or both skew-symmetry and sparsity [20] or the fact (47) may be regarded as the exponential map of a Lie algebra to a Lie group [13], but all of them require $O(n^3)$ cost. In [24], the exponential is computed via an SVD of B .

Fortunately for us, we have a fast algorithm for (47) based on Strang splitting [61] that takes time at most $12nk(n-k)$. First observe a matrix in the exponent of (47) may be written as a unique linear combination

$$\begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix} = \sum_{i=1}^k \sum_{j=1}^{n-k} \alpha_{ij} \begin{bmatrix} 0 & E_{ij} \\ -E_{ij}^\top & 0 \end{bmatrix} \quad (48)$$

where $\alpha_{ij} \in \mathbb{R}$ and E_{ij} is the matrix whose (i, j) entry is one and other entries are zero. Observe that

$$\exp\left(\theta \begin{bmatrix} 0 & E_{ij} \\ -E_{ij}^\top & 0 \end{bmatrix}\right) = \begin{bmatrix} I + (\cos \theta - 1)E_{ii} & (\sin \theta)E_{ij} \\ -(\sin \theta)E_{ji} & I + (\cos \theta - 1)E_{jj} \end{bmatrix} =: G_{i,j+k}(\theta)$$

is a Givens rotation in the i th and $(j+k)$ th plane of θ radians [31, p. 240]. Strang splitting, applied recursively to (48), then allows us to approximate

$$\exp\left(\begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix}\right) \approx G_{1,1+k}\left(\frac{1}{2}\alpha_{11}\right)G_{1,2+k}\left(\frac{1}{2}\alpha_{12}\right) \cdots G_{k,n-1}\left(\frac{1}{2}\alpha_{k,n-k-1}\right)G_{k,n}\left(\alpha_{k,n-k}\right) \\ G_{k,n-1}\left(\frac{1}{2}\alpha_{k,n-k-1}\right) \cdots G_{1,2+k}\left(\frac{1}{2}\alpha_{12}\right)G_{1,1+k}\left(\frac{1}{2}\alpha_{11}\right). \quad (49)$$

Computing the product in (49) is thus equivalent to computing a sequence of $2k(n-k) - 1$ Givens rotations, which takes time $12nk(n-k) - 6n$. For comparison, directly evaluating (47) via an SVD of B would have taken time $4k(n-k)^2 + 22k^3 + 2n^3$ (first two summands for SVD [31, p. 493], last summand for two matrix-matrix products).

The approximation in (49) requires that $\|B\|$ be sufficiently small [61]. But as gradient goes to zero when the iterates converge to a minimizer, $\|B\|$ will eventually be small enough for Strang approximation. We initialize our Riemannian algorithms with retraction algorithms, which do not require matrix exponential, i.e., run a few steps of a retraction algorithm to get close to a minimizer before switching to a Riemannian algorithm.

³The retraction based on Cayley transform in Proposition 6.6 may be viewed as a special case of the Padé approximation method in [12].

7.2. Retraction algorithms. In manifold optimization algorithms, an iterate is a point on a manifold and a search direction is a tangent vector at that point. Retraction algorithms rely on the retraction map R_Q for updating iterates and vector transport T_Q for updating search directions. Our interest in retraction algorithms is primarily to use them to initialize the Riemannian algorithms in the next section, and as such we limit ourselves to the least expensive ones.

A retraction-based steepest descent avoids even vector transport and takes the simple form

$$Q_{i+1} = R_{Q_i}(-\alpha_i \nabla f(Q_i)),$$

an analogue of the usual $x_{i+1} = x_i - \alpha_i \nabla f(x_i)$ in Euclidean space. As for our choice of retraction map, again computational costs dictate that we exclude the projection R_Q^E in Proposition 6.4 since it requires an EVD, and limit ourselves to the QR retraction R_Q^Q or Cayley retraction R_Q^C in Propositions 6.5 and 6.6 respectively. We present the latter in Algorithm 1 as an example.

We select our step size α_i using the well-known Barzilai–Borwein formula [9] but any line search procedure may be used instead. Recall that over Euclidean space, there are two choices for the Barzilai–Borwein step size:

$$\alpha_i = \frac{s_{i-1}^\top s_{i-1}}{(g_i - g_{i-1})^\top s_{i-1}}, \quad \alpha_i = \frac{(g_i - g_{i-1})^\top s_{i-1}}{(g_i - g_{i-1})^\top (g_i - g_{i-1})}, \quad (50)$$

where $s_{i-1} := x_i - x_{i-1}$. On a manifold M , the gradient $g_{i-1} \in \mathbb{T}_{x_{i-1}} M$ would have to be first parallel transported to $\mathbb{T}_{x_i} M$ and the step s_{i-1} would need to be replaced by a tangent vector in $\mathbb{T}_{x_{i-1}} M$ so that the exponential map $\exp_{x_{i-1}}(s_{i-1}) = x_i$. Upon applying this procedure, we obtain

$$\alpha_i = \frac{\text{tr}(S_{i-1}^\top S_{i-1})}{\text{tr}((G_i - G_{i-1})^\top S_i)}, \quad \alpha_i = \frac{\text{tr}((G_i - G_{i-1})^\top S_{i-1})}{\text{tr}((G_i - G_{i-1})^\top (G_i - G_{i-1}))}. \quad (51)$$

In other words, it is as if we have naively replaced the g_i and s_i in (50) by the effective gradient G_i and the effective step S_i . But (51) is indeed the correct Riemannian expressions for Barzilai–Borwein step size in the involution model — the parallel transport and exponential map have already been taken into account when we derive (51). This is an example of the extraordinary simplicity of the involution model that we mentioned earlier and will see again in Section 7.3.

Of the two expressions for α_i in (51), we chose the one on the right because our effective gradient G_i , which is computed directly, is expected to be slightly more accurate than our effective step size S_i , which is computed from G_i . Other more sophisticated retraction algorithms [3] can be readily created for the involution model using the explicit expressions derived in Section 6.

7.3. Riemannian algorithms. Riemannian algorithms, called “geometric algorithms” in [24], are true geometric analogues of those on Euclidean spaces — straight lines are replaced by geodesic curves, displacements by parallel transports, inner products by Riemannian metrics, gradients and Hessians by their Riemannian counterparts. Every operation in a Riemannian algorithm is intrinsic: iterates stay on the manifold, conjugate and search directions stay in tangent spaces, and there are no geometrically meaningless operations like adding a point to a tangent vector or subtracting tangent vectors from two different tangent spaces.

The involution model, like other models in [2, 24, 34], supplies a system of extrinsic coordinates that allow geometric objects and operations to be computed with standard numerical linear algebra but it offers a big advantage, namely, one can work entirely with the effective gradients and effective steps. For example, it looks as if parallel transport is missing from our Algorithms 2–5, but that is only because the expressions in the involution model can be simplified to an extent that gives such an illusion. Our parallel transport is effectively contained in the step where we update the eigenbasis V_i to V_{i+1} .

We begin with steepest descent in Algorithm 2, the simplest of our four Riemannian algorithms. As in the case of Algorithm 1, we will use Barzilai–Borwein step size but any line search procedure may be used to produce α_i . In this case, any conceivable line search procedure would have required

Algorithm 1 Steepest descent with Cayley retraction1: Initialize $Q_0 = V_0 I_{k,n-k} V_0^\top \in \text{Gr}(k, n)$.2: **for** $i = 0, 1, \dots$ **do**3: compute effective gradient G_i at Q_i

▷ entries * not needed

$$V_i^\top (f_{Q_i} + f_{Q_i}^\top) V_i = \begin{bmatrix} * & 2G_i \\ 2G_i^\top & * \end{bmatrix};$$

4: **if** $i = 0$ **then**5: initialize $S_0 = -G_0$, $\alpha_0 = 1$;6: **else**

7: compute Barzilai–Borwein step

▷ or get α_i from line search

$$\alpha_i = \text{tr}((G_i - G_{i-1})^\top S_{i-1}) / \text{tr}((G_i - G_{i-1})^\top (G_i - G_{i-1}));$$

$$S_i = -\alpha_i G_i;$$

8: **end if**

9: perform Cayley transform

$$C_i = \begin{bmatrix} I & S_i/4 \\ -S_i^\top/4 & I \end{bmatrix} \begin{bmatrix} I & -S_i/4 \\ S_i^\top/4 & I \end{bmatrix}^{-1};$$

10: update eigenbasis

▷ effective vector transport

$$V_{i+1} = V_i C_i;$$

11: update iterate

$$Q_{i+1} = V_{i+1} I_{k,n-k} V_{i+1}^\top;$$

12: **end for**

us to search over a geodesic curve and thus having to evaluate matrix exponential multiple times, using the Barzilai–Borwein step size circumvents this problem entirely.

Unlike its retraction-based counterpart in Algorithm 1, here the iterates descent along geodesic curves. Algorithm 1 may in fact be viewed as an approximation of Algorithm 2 where the matrix exponential in Step 9 is replaced with its first-order Padé approximation, i.e., a Cayley transform.

Newton method, shown in Algorithm 3, is straightforward with the computation of Newton step as in (41). In practice, instead of a direct evaluation of $H_Q \in \mathbb{R}^{k(n-k) \times k(n-k)}$ as in (40), we determine H_Q in a manner similar to Corollary 5.3. When regarded as a linear map $H_Q : \mathbb{T}_Q \text{Gr}(k, n) \rightarrow \mathbb{T}_Q \text{Gr}(k, n)$, its value on a basis vector X_{ij} in (37) is

$$H_Q(X_{ij}) = \frac{1}{4} V \begin{bmatrix} 0 & B_{ij} + AE_{ij} - E_{ij}C \\ (B_{ij} + AE_{ij} - E_{ij}C)^\top & 0 \end{bmatrix} V^\top, \quad (52)$$

where A, C are as in (38) and B_{ij} is given by

$$V^\top (f_{QQ}(X_{ij}) + f_{QQ}(X_{ij})^\top) V = \begin{bmatrix} * & B_{ij} \\ B_{ij}^\top & * \end{bmatrix},$$

for all $i = 1, \dots, k$, $j = 1, \dots, n - k$. Note that these computations can be performed completely in parallel — with $k(n - k)$ cores, entries of H_Q can be evaluated all at once.

Our conjugate gradient uses the Polak–Ribière formula [55] for conjugate step size; it is straightforward to replace that with the formulas of Dai–Yuan [19], Fletcher–Reeves [27], or Hestenes–Stiefel [35]. For easy reference:

$$\begin{aligned} \beta_i^{\text{PR}} &= \text{tr}(G_{i+1}^\top (G_{i+1} - G_i)) / \text{tr}(G_i^\top G_i), & \beta_i^{\text{HS}} &= -\text{tr}(G_{i+1}^\top (G_{i+1} - G_i)) / \text{tr}(P_i^\top (G_{i+1} - G_i)), \\ \beta_i^{\text{FR}} &= \text{tr}(G_{i+1}^\top G_{i+1}) / \text{tr}(G_i^\top G_i), & \beta_i^{\text{DY}} &= -\text{tr}(G_{i+1}^\top G_{i+1}) / \text{tr}(P_i^\top (G_{i+1} - G_i)). \end{aligned} \quad (53)$$

Algorithm 2 Steepest descent1: Initialize $Q_0 = V_0 I_{k,n-k} V_0^\top \in \text{Gr}(k, n)$.2: **for** $i = 0, 1, \dots$ **do**3: compute effective gradient G_i at Q_i

▷ entries * not needed

$$V_i^\top (f_{Q_i} + f_{Q_i}^\top) V_i = \begin{bmatrix} * & 2G_i \\ 2G_i^\top & * \end{bmatrix};$$

4: **if** $i = 0$ **then**5: initialize $S_0 = -G_0$, $\alpha_0 = 1$;6: **else**

7: compute Barzilai–Borwein step

▷ or get α_i from line search

$$\alpha_i = \text{tr}((G_i - G_{i-1})^\top S_{i-1}) / \text{tr}((G_i - G_{i-1})^\top (G_i - G_{i-1}));$$

$$S_i = -\alpha_i G_i;$$

8: **end if**

9: update eigenbasis

▷ effective parallel transport

$$V_{i+1} = V_i \exp \left(\begin{bmatrix} 0 & -S_i/2 \\ S_i^\top/2 & 0 \end{bmatrix} \right);$$

10: update iterate

$$Q_{i+1} = V_{i+1} I_{k,n-k} V_{i+1}^\top;$$

11: **end for****Algorithm 3** Newton's method1: Initialize $Q_0 = V_0 I_{k,n-k} V_0^\top \in \text{Gr}(k, n)$.2: **for** $i = 0, 1, \dots$ **do**3: compute effective gradient G_i at Q_i

$$V_i^\top (f_{Q_i} + f_{Q_i}^\top) V_i = \begin{bmatrix} A_i & 2G_i \\ 2G_i^\top & C_i \end{bmatrix};$$

4: generate Hessian matrix H_Q by (40) or (52);5: solve for effective Newton step S_i

$$H_Q \text{vec}(S_i) = -\text{vec}(G_i);$$

6: update eigenbasis

▷ effective parallel transport

$$V_{i+1} = V_i \exp \left(\begin{bmatrix} 0 & S_i/2 \\ -S_i^\top/2 & 0 \end{bmatrix} \right);$$

7: update iterate

$$Q_{i+1} = V_{i+1} I_{k,n-k} V_{i+1}^\top;$$

8: **end for**

It may appear from these formulas that we are subtracting tangent vectors from tangent spaces at different points but this is an illusion. The effective gradients G_i and G_{i+1} are defined by the Riemannian gradients $\nabla f(Q_i) \in \mathbb{T}_{Q_i} \text{Gr}(k, n)$ and $\nabla f(Q_{i+1}) \in \mathbb{T}_{Q_{i+1}} \text{Gr}(k, n)$ as in (39) but they are not Riemannian gradients themselves. The formulas in (53) have in fact already accounted for the requisite parallel transports. This is another instance of the simplicity afforded by the involution model that we saw earlier in our Barzilai–Borwein step size (51) — our formulas in (53) are no different from the standard formulas for Euclidean space in [19, 27, 35, 55]. Contrast these with

the formulas in [24, Equations 2.80 and 2.81], where the parallel transport operator τ makes an explicit appearance and cannot be avoided.

Algorithm 4 Conjugate gradient

1: Initialize $Q_0 = V_0 I_{k,n-k} V_0^\top \in \text{Gr}(k, n)$.

2: Compute effective gradient G_0 at Q_0

▷ entries * not needed

$$V_0^\top (f_{Q_0} + f_{Q_0}^\top) V_0 = \begin{bmatrix} * & 2G_0 \\ 2G_0^\top & * \end{bmatrix};$$

3: initialize $P_0 = S_0 = -G_0$, $\alpha_0 = 1$;

4: **for** $i = 0, 1, \dots$ **do**

5: compute α_i from line search of direction P_i and set

$$S_i = \alpha_i P_i;$$

6: update eigenbasis

▷ effective parallel transport

$$V_{i+1} = V_i \exp \left(\begin{bmatrix} 0 & -S_i/2 \\ S_i^\top/2 & 0 \end{bmatrix} \right);$$

7: update iterate

$$Q_{i+1} = V_{i+1} I_{k,n-k} V_{i+1}^\top;$$

8: compute effective gradient G_{i+1} at Q_{i+1}

▷ entries * not needed

$$V_{i+1}^\top (f_X(Q_{i+1}) + f_X(Q_{i+1})^\top) V_{i+1} = \begin{bmatrix} * & 2G_{i+1} \\ 2G_{i+1}^\top & * \end{bmatrix};$$

9: compute Polak–Ribière conjugate step size

$$\beta_i = \text{tr}((G_{i+1} - G_i)^\top G_{i+1}) / \text{tr}(G_i^\top G_i);$$

10: update conjugate direction

$$P_{i+1} = -G_{i+1} + \beta_i P_i;$$

11: **end for**

Our quasi-Newton method, given in Algorithm 5, uses L-BFGS updates with two loops recursion [54]. Observe that a minor feature of Algorithms 1, 2, 4, 5 is that they do not require vectorization of matrices; everything can be computed in terms of matrix-matrix products, allowing for Strassen-style fast algorithms. While it is straightforward to replace the L-BFGS updates with full BFGS, DFP, SR1, or Broyden class updates, doing so will require that we vectorize matrices like in Algorithm 3.

7.4. Exponential-free algorithms? This brief section is speculative and may be safely skipped. In our algorithms, an exponential matrix $U := \exp(\begin{bmatrix} 0 & B \\ -B^\top & 0 \end{bmatrix})$ is always⁴ applied as a conjugation of some *symmetric* matrix $X \in \mathbb{R}^{n \times n}$:

$$X \mapsto UXU^\top \quad \text{or} \quad X \mapsto U^\top XU. \tag{54}$$

In other words, the Givens rotations in (49) are applied in the form of *Jacobi rotations* [31, p. 477]. For a symmetric X , a Jacobi rotation $X \mapsto G_{ij}(\theta) X G_{ij}(\theta)^\top$ takes the same number (as opposed to twice the number) of floating point operations as a Givens rotation applied on the left, $X \mapsto G_{ij}(\theta) X$, or on the right, $X \mapsto X G_{ij}(\theta)$. Thus with Strang splitting the operations in (54) take time $12nk(n-k)$. To keep our algorithms simple, we did not take advantage of this observation.

In principle, one may avoid any actual computation of matrix exponential by simply storing the $k(n-k)$ Givens rotations in (49) without actually forming the product, and apply them as Jacobi rotations whenever necessary. The storage of $G_{ij}(\theta)$ requires just a single floating point number θ

⁴See steps 3, 10 in Algorithm 2; steps 3, 7 in Algorithm 3; steps 7, 8 in Algorithm 4; steps 3, 20 in Algorithm 5.

Algorithm 5 Quasi-Newton with L-BFGS updates

```

1: Initialize  $Q_0 = V_0 I_{k,n-k} V_0^\top \in \text{Gr}(k, n)$ .
2: for  $i = 0, 1, \dots$  do
3:   Compute effective gradient  $G_i$  at  $Q_i$  ▷ entries * not needed

$$V_i^\top (f_X(Q_i) + f_X(Q_i)^\top) V_i = \begin{bmatrix} * & 2G_i \\ 2G_i^\top & * \end{bmatrix};$$

4:   if  $i = 0$  then
5:     initialize  $S_0 = -G_0$ ;
6:   else
7:     set  $Y_{i-1} = G_i - G_{i-1}$  and  $P = G_i$ ; ▷  $P$  is temporary variable for loop
8:     for  $j = i - 1, \dots, \max(0, i - m)$  do
9:        $\alpha_j = \text{tr}(S_j^\top P) / \text{tr}(Y_j^\top S_j)$ ;
10:       $P = P - \alpha_j Y_j$ ;
11:     end for
12:     set  $Z = \text{tr}(Y_{i-1}^\top S_{i-1}) / \text{tr}(Y_{i-1}^\top Y_{i-1}) P$ ; ▷  $Z$  is temporary variable for loop
13:     for  $j = \max(0, i - m), \dots, i - 1$  do
14:        $\beta_j = \text{tr}(Y_j^\top Z) / \text{tr}(Y_j^\top S_j)$ ;
15:        $Z = Z + (\alpha_j - \beta_j) S_j$ ;
16:     end for
17:     set effective quasi-Newton step  $S_i = -Z$ ;
18:   end if
19:   update eigenbasis ▷ effective parallel transport

$$V_{i+1} = V_i \exp \left( \begin{bmatrix} 0 & -S_i/2 \\ S_i^\top/2 & 0 \end{bmatrix} \right);$$

20:   update iterate

$$Q_{i+1} = V_{i+1} I_{k,n-k} V_{i+1}^\top;$$

21: end for

```

and two indices but one would need to figure out how to update these $k(n - k)$ Givens rotations from one iteration to the next. We leave this as an open problem for interested readers.

7.5. Drawbacks of quotient models. This section, which may again be safely skipped, discusses the pitfalls of modeling a manifold as a homogeneous space of matrices. In our context, this would be the orthogonal, Stiefel, and full-rank models:

$$\text{Gr}(k, n) \cong \text{O}(n) / (\text{O}(n - k) \times \text{O}(k)) \cong \text{V}(k, n) / \text{O}(k) \cong \mathbb{R}_k^{n \times k} / \text{GL}(k). \quad (55)$$

Such homogeneous space models take the form of a quotient $B = E/G$ where B is the manifold we want to optimize over, E is some other manifold on which we have optimization algorithms, and G is some Lie group. The quotient map $\pi : E \rightarrow B$, $x \mapsto \llbracket x \rrbracket$ defines a *principal bundle*. The idea of Riemannian optimization algorithms for such models is to lift every point on B up to the total space E so that optimization algorithms on E can be applied. In Section 1, we only mentioned the computational costs that come with lifting a point $\llbracket x \rrbracket \in B$ to $x \in E$ and with checking equality of points $\llbracket x_1 \rrbracket = \llbracket x_2 \rrbracket$ given x_1, x_2 . Here we focus on a more serious mathematical difficulty.

Since our goal is optimization, we cannot simply lift points on B to E in arbitrary ways. Ideally, whatever method of lifting should at least be continuous, i.e., nearby points in B are lifted to nearby points in E . In fact if we need first or second derivatives for the purpose of optimization, then the lifting has to be differentiable to first or second order, as we will see below. In differential geometric lingo, finding a lifting j is called finding a *global section* and it is impossible for any of the models

in (55). Take the Stiefel model for illustration, the quotient map $\pi : V(k, n) \rightarrow V(k, n)/O(k)$, $Y \mapsto \llbracket Y \rrbracket$, defines $\text{Gr}(k, n)$ as an $O(k)$ -principal bundle. This is not a trivial bundle, which is equivalent to π not admitting a global section [38]. The consequence is that there is no global ‘Stiefel coordinates’ for $\text{Gr}(k, n)$, i.e., we cannot represent all points of $\text{Gr}(k, n)$ by points of $V(k, n)$ in a continuous manner.

We will use the Stiefel model as an illustration of the above discussion. Let $f : V(k, n)/O(k) \rightarrow \mathbb{R}$ be a differentiable objective function and let $\pi : V(k, n) \rightarrow V(k, n)/O(k)$ be the quotient map, which is always smooth. A lifting is a right inverse $j : V(k, n)/O(k) \rightarrow V(k, n)$ to π , i.e., $\pi \circ j(\llbracket Y \rrbracket) = \llbracket Y \rrbracket$ for any $\llbracket Y \rrbracket \in V(k, n)/O(k)$. Note that such a map j is not unique. In order to use standard numerical linear algebra, we have to work with actual matrices in $V(k, n)$ as opposed to equivalence classes of matrices in $V(k, n)/O(k)$, so we need a way to assign to any equivalence class in $V(k, n)/O(k)$ a matrix in $V(k, n)$ that represents that equivalence class — any such assignment gives us a lifting.

Upon selecting a lifting j , we may then transform the problem of optimizing $f : V(k, n)/O(k) \rightarrow \mathbb{R}$ to optimizing $f \circ \pi : j(V(k, n)/O(k)) \rightarrow \mathbb{R}$. Note that $j(V(k, n)/O(k)) \subseteq V(k, n)$ is a set of actual matrices. The issue is that in order to have gradients and Hessians we also need $f \circ \pi$ to be a differentiable function — this is only possible if its domain $j(V(k, n)/O(k))$ is a differential manifold. But as we saw, we cannot even choose j to be continuous, so $j(V(k, n)/O(k))$ is not even a topological manifold. In the involution and projection models, we do not face these issues as a point in $\text{Gr}(k, n)$ is already represented by a matrix.

8. NUMERICAL EXPERIMENTS

We will describe three sets of numerical experiments, testing Algorithms 1–5 on three different objective functions, the first two are chosen because their true solutions can be independently determined in closed-form, allowing us to ascertain that our algorithms have converged to the global optimizer. All our codes are open source and publicly available at:

<https://github.com/laizehua/Simpler-Grassmannians>

The goal of these numerical experiments is to compare our algorithms for the involution model in Section 7 with the corresponding algorithms for the Stiefel model in [24]. Algorithm 5, although implemented in our codes, is omitted from our comparisons as quasi-Newton methods are not found in [24].

8.1. Quadratic function. The standard test function for Grassmannian optimization is the quadratic form in [24, Section 4.4] which, in the Stiefel model, takes the form $\text{tr}(Y^T F Y)$ for a symmetric $F \in \mathbb{R}^{n \times n}$ and $Y \in V(k, n)$. By Proposition 2.4, we write $Q = 2Y Y^T - I$, then $\text{tr}(Y^T F Y) = (\text{tr}(F Q) + \text{tr}(F))/2$. Therefore, in the involution model, this optimization problem takes an even simpler form

$$f(Q) = \text{tr}(F Q) \tag{56}$$

for $Q \in \text{Gr}(k, n)$. What was originally quadratic in the Stiefel model becomes linear in the involution model. The minimizer of f ,

$$Q_* := \text{argmin}\{\text{tr}(F Q) : Q^T Q = I, Q^T = Q, \text{tr}(Q) = 2k - n\},$$

is given by $Q_* = \Pi V I_{k, n-k} V^T \Pi^T$ where

$$\Pi = \begin{bmatrix} & & & 1 \\ & \ddots & & \\ & & & \\ 1 & & & \end{bmatrix} \quad \text{and} \quad \frac{F + F^T}{2} = V D V^T$$

is an eigendecomposition with eigenbasis $V \in O(n)$ and eigenvalues $D := \text{diag}(\lambda_1, \dots, \lambda_n)$ in descending order. This follows from essentially the same argument⁵ used in the proof of Lemma 6.3 and the corresponding minimum is $f(Q_*) = -\lambda_1 - \dots - \lambda_k + \lambda_{k+1} + \dots + \lambda_n$.

For the function $f(Q) = \text{tr}(FQ)$, the effective gradient $G_i \in \mathbb{R}^{k \times (n-k)}$ in Algorithms 2, 4, 5 at the point $Q_i = V_i I_{k, n-k} V_i^\top \in \text{Gr}(k, n)$ is given by

$$V_i^\top F V_i = \begin{bmatrix} A & G_i \\ G_i^\top & C \end{bmatrix}.$$

The matrices $A \in \mathbb{R}^{k \times k}$ and $C \in \mathbb{R}^{(n-k) \times (n-k)}$ are not needed for Algorithms 2, 4, 5 but they are required in Algorithm 3. Indeed, the effective Newton step $S_i \in \mathbb{R}^{k \times (n-k)}$ in Algorithm 3 is obtained by solving the Sylvester equation

$$A S_i - S_i C = 2 G_i.$$

To see this, note that by Proposition 5.2, for any $B \in \mathbb{R}^{k \times (n-k)}$,

$$\begin{aligned} \nabla^2 f(Q_i) \left(V_i \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} V_i^\top, V_i \begin{bmatrix} 0 & S_i \\ S_i^\top & 0 \end{bmatrix} V_i^\top \right) &= -\frac{1}{2} \text{tr} \left(\begin{bmatrix} A & G_i \\ G_i^\top & C \end{bmatrix} \begin{bmatrix} X S_i^\top + S_i B^\top & 0 \\ 0 & -B^\top S_i - S_i^\top B \end{bmatrix} \right) \\ &= -\text{tr}(B^\top (A S_i - S_i C)), \end{aligned}$$

and to obtain the effective Newton step (41), we simply set the last term to be equal to $-2 \text{tr}(B^\top G_i)$.

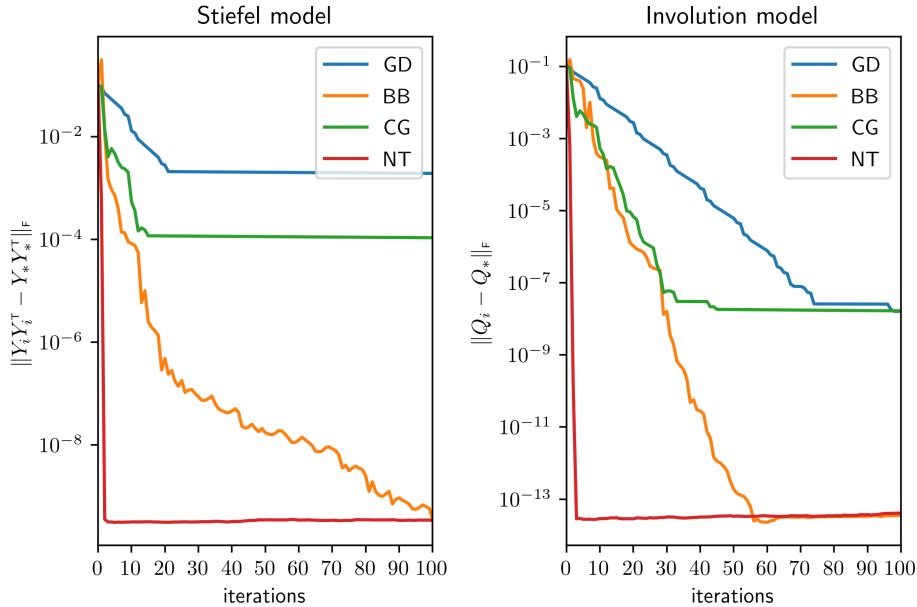


FIGURE 1. Convergence behavior of algorithms in the Stiefel and involution models.

Figure 1 compares the convergence behaviors of the algorithms in [24] for the Stiefel model and our Algorithms 2, 3, 4 in the involution model: steepest descent with line search (GD) and with Barzilai–Borwein step size (BB), conjugate gradient (CG), and Newton’s method (NT) for $k = 6$, $n = 16$. We denote the i th iterate in the Stiefel and involution models by Y_i and Q_i respectively — note that Y_i is a 16×6 matrix with orthonormal columns whereas Q_i is a 16×16 symmetric orthogonal matrix. All algorithms are fed the same initial point obtained from 20 iterations of Algorithm 1. The matrix F is generated randomly with standard normal entries. We use the

⁵Recall also that for any real numbers $a_1 \leq \dots \leq a_n$, $b_1 \leq \dots \leq b_n$, and any permutation π , one always have that $a_1 b_n + a_2 b_{n-1} + \dots + a_n b_1 \leq a_1 b_{\pi(1)} + a_2 b_{\pi(2)} + \dots + a_n b_{\pi(n)} \leq a_1 b_1 + a_2 b_2 + \dots + a_n b_n$.

Armijo conditions for linesearch in GD and CG; but as in the Euclidean case, BB and NT are used as is without linesearch. Since we have the true global minimizer in closed form, denoted by Y_* and Q_* in the respective model, the error is given by geodesic distance to the true solution. For convenience we compute $\|Y_i Y_i^T - Y_* Y_*^T\|_F$ and $\|Q_i - Q_*\|_F$, which are constant multiples of the chordal distance [68, Table 2] (also called projection F-norm [24, p. 337]) and are equivalent, in the sense of metrics, to the geodesic distance. Since we use a log scale, the vertical axes of the two graphs in Figure 1 are effectively both geodesic distance and, in particular, their values may be compared. The conclusion is clear: While Algorithms 2 (BB) and 3 (NT) in the involution model attain a level of accuracy on the order of machine precision, the corresponding algorithms in the Stiefel model do not. The reason is numerical stability, as we will see next.

Figure 2 shows the loss of orthogonality for various algorithms in the Stiefel and involution models, measured respectively by $\|Y_i^T Y_i - I\|_F$ and $\|Q_i^2 - I\|_F$. In the Stiefel model, the deviation from orthogonality $\|Y_i^T Y_i - I\|_F$ grows exponentially. In the worst case, the GD iterates Y_i , which of course ought to be of rank $k = 6$, actually converged to a rank-one matrix. In the involution model, the deviation from orthogonality $\|Q_i^2 - I\|_F$ remains below 10^{-13} for all algorithms — the loss-of-orthogonality is barely noticeable.

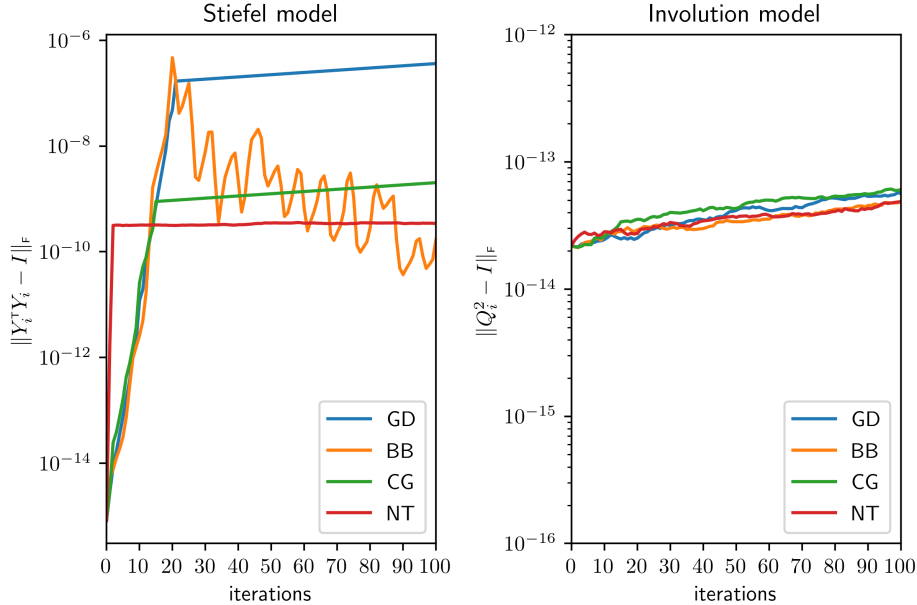


FIGURE 2. Loss of orthogonality in the Stiefel and involution models.

A closer inspection of the algorithms for NT [24, p. 325] and CG [24, p. 327] in the Stiefel model reveals why: A point Y_i and the gradient G_i at that point are highly dependent on each other — an ε -deviation from orthogonality in Y_i results in an ε -error in G_i that in turn becomes a 2ε -deviation from orthogonality in Y_{i+1} , i.e., one loses orthogonality at an exponential rate. We may of course reorthogonalize Y_i at every iteration in the Stiefel model to artificially enforce the orthonormality of its columns but this incurs additional cost and turns a Riemannian algorithm into a retraction algorithm, as reorthogonalization of Y_i is effectively a QR retraction.

Contrast this with the involution model: In Algorithms 3 (NT) and 4 (CG), the point Q_i and the effective gradient G_i are both computed directly from the eigenbasis V_i , which is updated to V_{i+1} by an orthogonal matrix, or a sequence of Givens rotations if one uses Strang splitting as in (49). This introduces a small (constant order) deviation from orthogonality each step. Consequently, the deviation from orthogonality at worst grows linearly.

8.2. Grassmann Procrustes problem. Let $k, m, n \in \mathbb{N}$ with $k \leq n$. Let $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times k}$. The minimization problem

$$\min_{Q^T Q = I} \|AQ - B\|_F,$$

is called the Stiefel Procrustes problem [24, Section 3.5.2] and the special case $k = n$ is the usual orthogonal Procrustes problem [31, Section 6.4.1]. Respectively, these are

$$\min_{Q \in V(k, n)} \|AQ - B\|_F \quad \text{and} \quad \min_{Q \in O(n)} \|AQ - B\|_F.$$

One might perhaps wonder if there is also a *Grassmann Procrustes problem*

$$\min_{Q \in \text{Gr}(k, n)} \|AQ - B\|_F. \tag{57}$$

In fact, with the involution model for $\text{Gr}(k, n)$, the problem (57) makes perfect sense, with the requirement that $k = n$. The same argument in the proof of Lemma 6.3 shows that the minimizer Q_* of (57) is given by $Q_* = VI_{k, n-k}V^T$ where

$$\frac{A^T B + B^T A}{2} = V D V^T$$

is an eigendecomposition with eigenbasis $V \in O(n)$ and eigenvalues $D := \text{diag}(\lambda_1, \dots, \lambda_n)$ in descending order. The convergence and loss-of-orthogonality behaviors for this problem are very similar to those in Section 8.1 and provides further confirmation for the earlier numerical results. The plots from solving (57) for arbitrary A, B using any of Algorithms 2–5 are generated in our codes but as they are nearly identical to Figures 1 and 2 we omit them here.

8.3. Fréchet mean and Karcher mean. Let $Q_1, \dots, Q_m \in \text{Gr}(k, n)$ and consider the sum-of-square-distances minimization problem:

$$\min_{Q \in \text{Gr}(k, n)} \sum_{j=1}^m d^2(Q_j, Q), \tag{58}$$

where d is the geodesic distance in (31). The global minimizer of this problem is called the *Fréchet mean* and a local minimizer is called a *Karcher mean* [43]. For the case $m = 2$, a Fréchet mean is the midpoint, i.e., $t = 1/2$, of the geodesic connecting Q_1 and Q_2 given by the closed-form expression in Proposition 4.5. The objective function f in (58) is differentiable almost everywhere⁶ with its Riemannian gradient [42] given by

$$\nabla f(Q) = 2 \sum_{j=1}^m \log_Q(Q_j),$$

where the logarithmic map is as in Corollary 4.7. To the best of our knowledge, there is no simple expression for $\nabla^2 f(Q)$ and as such we exclude Newton method from consideration below.

We will set $k = 6$, $n = 16$, and $m = 3$. Unlike the problems in Sections 8.1 and 8.2, the problem in (58) does not have a closed-form solution when $m > 2$. Consequently we quantify convergence behavior in Figure 3 by the rate gradient goes to zero. The deviation from orthogonality is quantified as in Section 8.1 and shown in Figure 4. The instability of the algorithms in the Stiefel model is considerably more pronounced here — both GD and CG failed to converge to a stationary point as we see in Figure 3. The cause, as revealed by Figure 4, is a severe loss-of-orthogonality that we will elaborate below.

The expression for geodesic distance $d(Y, Y')$ between two points Y, Y' in the Stiefel model (see [2, Section 3.8] or [68, Equation 7]) is predicated on the crucial assumption that each of these matrices has orthonormal columns. As a result, a moderate deviation from orthonormality in an

⁶ f is nondifferentiable only when Q falls on the cut locus of Q_i for some i but the union of all cut loci of Q_1, \dots, Q_m has codimension ≥ 1 .

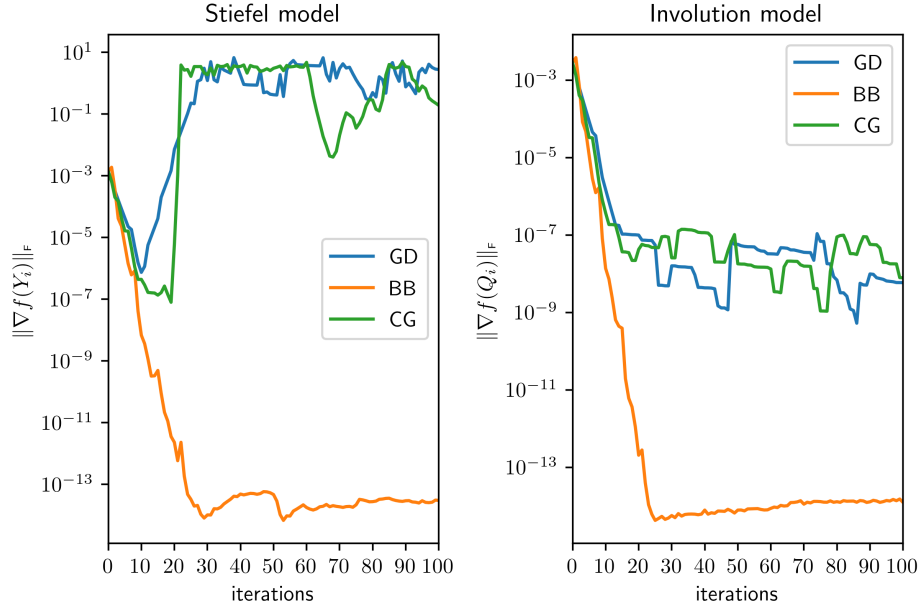


FIGURE 3. Convergence behavior of algorithms in the Stiefel and involution models.

iterate Y leads to vastly inaccurate values in the objective function value $f(Y)$, which is a sum of m geodesic distances *squared*. This is reflected in the graphs on the left of Figure 3 for the GD and CG algorithms, whose step sizes come from line search and depend on these function values. Using the BB step size, which does not depend on objective function values, avoids the issue. But for GD and CG, the reliance on inaccurate function values leads to further loss-of-orthogonality, and when the columns of an iterate Y are far from orthonormal, plugging Y into the expression for gradient simply yields a nonsensical result, at times even giving an *ascent* direction in a minimization problem.

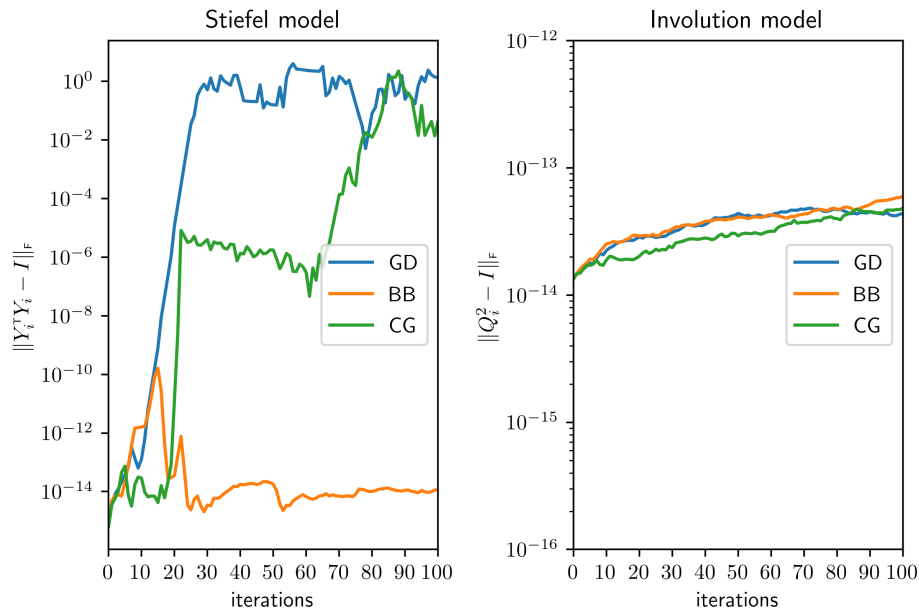


FIGURE 4. Loss of orthogonality in the Stiefel and involution models.

For all three algorithms in the involution model, the deviation from orthogonality in the iterates is kept at a negligible level of under 10^{-13} over the course of 100 iterations.

Acknowledgment. We would like to acknowledge the intellectual debt we owe to [2, 24, 34]. The work in this article would not have been possible without drawing from their prior investigations. The authors would also like to thank Nicolas Boumal for many helpful comments.

ZL and LHL are supported by the DARPA grant HR00112190040. In addition, ZL is supported by a Neubauer Family Distinguished Doctoral Fellowship from the University of Chicago, and LHL is supported by NSF IIS 1546413, DMS 1854831, and the Eckhardt Faculty Fund. KY is supported by NSFC Grant no. 11688101, NSFC Grant no. 11801548 and National Key R&D Program of China Grant no. 2018YFA0306702.

REFERENCES

- [1] H. Abe and T. Matsumura. Schur polynomials and weighted Grassmannians. *J. Algebraic Combin.*, 42(3):875–892, 2015.
- [2] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Appl. Math.*, 80(2):199–220, 2004.
- [3] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008.
- [4] P.-A. Absil and J. Malick. Projection-like retractions on matrix manifolds. *SIAM J. Optim.*, 22(1):135–158, 2012.
- [5] R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub. Newton’s method on Riemannian manifolds and a geometric model for the human spine. *IMA J. Numer. Anal.*, 22(3):359–390, 2002.
- [6] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. D. J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users’ Guide*. SIAM, Philadelphia, PA, third edition, 1999.
- [7] C. Bachoc, R. Coulangeon, and G. Nebe. Designs in Grassmannian spaces and lattices. *J. Algebraic Combin.*, 16(1):5–19, 2002.
- [8] A. Barg and D. Y. Nogin. Bounds on packings of spheres in the Grassmann manifold. *IEEE Trans. Inform. Theory*, 48(9):2450–2454, 2002.
- [9] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988.
- [10] R. Bhatia. Linear algebra to quantum cohomology: the story of Alfred Horn’s inequalities. *Amer. Math. Monthly*, 108(4):289–318, 2001.
- [11] A. R. Calderbank, R. H. Hardin, E. M. Rains, P. W. Shor, and N. J. A. Sloane. A group-theoretic framework for the construction of packings in Grassmannian spaces. *J. Algebraic Combin.*, 9(2):129–140, 1999.
- [12] J. R. Cardoso and F. S. Leite. Exponentials of skew-symmetric matrices and logarithms of orthogonal matrices. *J. Comput. Appl. Math.*, 233(11):2867–2875, 2010.
- [13] E. Celledoni and A. Iserles. Methods for the approximation of the matrix exponential in a Lie-algebraic setting. *IMA J. Numer. Anal.*, 21(2):463–488, 2001.
- [14] J. Cheeger and D. G. Ebin. *Comparison theorems in Riemannian geometry*. AMS Chelsea Publishing, Providence, RI, 2008.
- [15] Y. Chikuse. *Statistics on special manifolds*, volume 174 of *Lecture Notes in Statistics*. Springer-Verlag, New York, NY, 2003.
- [16] E. S. Coakley, V. Rokhlin, and M. Tygert. A fast randomized algorithm for orthogonal projection. *SIAM J. Sci. Comput.*, 33(2):849–868, 2011.
- [17] J. H. Conway, R. H. Hardin, and N. J. A. Sloane. Packing lines, planes, etc.: packings in Grassmannian spaces. *Experiment. Math.*, 5(2):139–159, 1996.
- [18] J. X. da Cruz Neto, L. L. de Lima, and P. R. Oliveira. Geodesic algorithms in Riemannian geometry. *Balkan J. Geom. Appl.*, 3(2):89–100, 1998.
- [19] Y. H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.*, 10(1):177–182, 1999.
- [20] N. Del Buono, L. Lopez, and R. Peluso. Computation of the exponential of large sparse skew-symmetric matrices. *SIAM J. Sci. Comput.*, 27(1):278–293, 2005.
- [21] J. W. Demmel. *Applied numerical linear algebra*. SIAM, Philadelphia, PA, 1997.
- [22] M. P. do Carmo. *Riemannian geometry*. Mathematics: Theory & Applications. Birkhäuser, Boston, MA, 1992.
- [23] D. W. Dreisigmeyer. Direct search methods on reductive homogeneous spaces. *J. Optim. Theory Appl.*, 176(3):585–604, 2018.
- [24] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1999.

- [25] M. Ehler and M. Gräf. Reproducing kernels for the irreducible components of polynomial spaces on unions of Grassmannians. *Constr. Approx.*, 49(1):29–58, 2019.
- [26] M. Farber and A. Postnikov. Arrangements of equal minors in the positive Grassmannian. *Adv. Math.*, 300:788–834, 2016.
- [27] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Comput. J.*, 7:149–154, 1964.
- [28] P. Galashin and P. Pylyavskyy. Ising model and the positive orthogonal Grassmannian. *Duke Math. J.*, 169(10):1877–1942, 2020.
- [29] E. S. Gawlik, Y. Nakatsukasa, and B. D. Sutton. A backward stable algorithm for computing the CS decomposition via the polar decomposition. *SIAM J. Matrix Anal. Appl.*, 39(3):1448–1469, 2018.
- [30] D. G. Giovanis and M. D. Shields. Data-driven surrogates for high dimensional models using Gaussian process regression on the Grassmann manifold. *Comput. Methods Appl. Mech. Engrg.*, 370:113269, 2020.
- [31] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.
- [32] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson. Extrinsic methods for coding and dictionary learning on Grassmann manifolds. *Int. J. Comput. Vis.*, 114(2-3):113–136, 2015.
- [33] S. Helgason. *Differential geometry, Lie groups, and symmetric spaces*, volume 34 of *Graduate Studies in Mathematics*. AMS, Providence, RI, 2001.
- [34] U. Helmke, K. Hüper, and J. Trumpf. Newton’s method on Grassmann manifolds. *preprint*, arXiv:0709.2205, 2007.
- [35] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 49:409–436 (1953), 1952.
- [36] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, Philadelphia, PA, second edition, 2002.
- [37] N. J. Higham. *Functions of matrices*. SIAM, Philadelphia, PA, 2008.
- [38] D. Husemoller. *Fibre bundles*, volume 20 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, third edition, 1994.
- [39] B. Jiang and Y.-H. Dai. A framework of constraint preserving update schemes for optimization on Stiefel manifold. *Math. Program.*, 153(2, Ser. A):535–575, 2015.
- [40] J. Jost. *Riemannian geometry and geometric analysis*. Universitext. Springer, Cham, seventh edition, 2017.
- [41] V. Jurdjevic, I. Markina, and F. Silva Leite. Extremal curves on Stiefel and Grassmann manifolds. *J. Geom. Anal.*, 108(4):289–318, 2019.
- [42] H. Karcher. Riemannian center of mass and mollifier smoothing. *Comm. Pure Appl. Math.*, 30(5):509–541, 1977.
- [43] H. Karcher. Riemannian center of mass and so called Karcher mean. *preprint*, arXiv:1407.2087, 2014.
- [44] S. N. Karp. Sign variation, the Grassmannian, and total positivity. *J. Combin. Theory Ser. A*, 145:308–339, 2017.
- [45] T. Kato. *Perturbation theory for linear operators*. Classics in Mathematics. Springer-Verlag, Berlin, 1995.
- [46] S. Kobayashi and K. Nomizu. *Foundations of differential geometry. Vol. II*. Wiley Classics Library. John Wiley and Sons, New York, NY, 1996.
- [47] I. Le and C. Fraser. Tropicalization of positive Grassmannians. *Selecta Math. (N.S.)*, 25(5):Paper No. 75, 55 pp., 2019.
- [48] P. Mattila. *Geometry of sets and measures in Euclidean spaces*, volume 44 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1995.
- [49] A. Medra and T. N. Davidson. Incremental Grassmannian feedback schemes for multi-user MIMO systems. *IEEE Trans. Signal Process.*, 63(5):1130–1143, 2015.
- [50] B. Mishra, H. Kasai, P. Javanpuria, and A. Saroop. A Riemannian gossip approach to subspace learning on Grassmann manifold. *Mach. Learn.*, 108(10):1783–1803, 2019.
- [51] B. Mishra and R. Sepulchre. Riemannian preconditioning. *SIAM J. Optim.*, 26(1):635–660, 2016.
- [52] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.*, 45(1):3–49, 2003.
- [53] L. I. Nicolaescu. *Lectures on the geometry of manifolds*. World Scientific, Hackensack, NJ, second edition, 2007.
- [54] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, second edition, 2006.
- [55] E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *Rev. Française Informat. Recherche Opérationnelle*, 3(16):35–43, 1969.
- [56] A. Sarlette and R. Sepulchre. Consensus optimization on manifolds. *SIAM J. Control Optim.*, 48(1):56–76, 2009.
- [57] M. Shub. Some remarks on dynamical systems and numerical analysis. In *Dynamical systems and partial differential equations (Caracas, 1984)*, pages 69–91. Univ. Simon Bolivar, Caracas, 1986.
- [58] G. Sonnevend, J. Stoer, and G. Zhao. On the complexity of following the central path of linear programs by linear extrapolation. II. volume 52, pages 527–553. 1991.

- [59] G. W. Stewart. Computing the CS decomposition of a partitioned orthonormal matrix. *Numer. Math.*, 40(3):297–306, 1982.
- [60] G. W. Stewart. *Matrix algorithms I: Basic decompositions*. SIAM, Philadelphia, PA, 1998.
- [61] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.
- [62] Y. Sun, N. Flammarion, and M. Fazel. Escaping from saddle points on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, pages 7276–7286, 2019.
- [63] L. N. Trefethen and D. Bau, III. *Numerical linear algebra*. SIAM, Philadelphia, PA, 1997.
- [64] D. S. Watkins. *Fundamentals of matrix computations*. Pure and Applied Mathematics. John Wiley and Sons, Hoboken, NJ, third edition, 2010.
- [65] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Math. Program.*, 142(1-2, Ser. A):397–434, 2013.
- [66] J. H. Wilkinson. *The algebraic eigenvalue problem*. Monographs on Numerical Analysis. Oxford University Press, New York, NY, 1988.
- [67] Y.-C. Wong. Differential geometry of Grassmann manifolds. *Proc. Nat. Acad. Sci. U.S.A.*, 57:589–594, 1967.
- [68] K. Ye and L.-H. Lim. Schubert varieties and distances between subspaces of different dimensions. *SIAM J. Matrix Anal. Appl.*, 37(3):1176–1197, 2016.
- [69] G. Zhao. Representing the space of linear programs as the Grassmann manifold. *Math. Program.*, 121(2, Ser. A):353–386, 2010.
- [70] L. Zheng and D. N. C. Tse. Communication on the Grassmann manifold: a geometric approach to the noncoherent multiple-antenna channel. *IEEE Trans. Inform. Theory*, 48(2):359–383, 2002.
- [71] W. Ziller. Examples of Riemannian manifolds with non-negative sectional curvature. In *Surveys in differential geometry. Vol. XI*, volume 11 of *Surv. Differ. Geom.*, pages 63–102. International Press, Somerville, MA, 2007.

COMPUTATIONAL AND APPLIED MATHEMATICS INITIATIVE, UNIVERSITY OF CHICAGO, CHICAGO, IL 60637-1514.
Email address: laizehua@uchicago.edu, lekheng@uchicago.edu

KLMM, ACADEMY OF MATHEMATICS AND SYSTEMS SCIENCE, CHINESE ACADEMY OF SCIENCES, BEIJING 100190,
CHINA
Email address: keyk@amss.ac.cn