Electronic Transactions on Numerical Analysis.
Volume 60, pp. 501–519, 2024.

ISSN 1068–9613.
DOI: 10.1553/etna_vol60s501

ETNA
Kent State University and
Johann Radon Institute (RICAM)

A SINGLE SHOOTING METHOD WITH APPROXIMATE FRÉCHET DERIVATIVE FOR COMPUTING GEODESICS ON THE STIEFEL MANIFOLD*

MARCO SUTTI†

Abstract. This paper shows how to use the shooting method, a classical numerical algorithm for solving boundary value problems, to compute the Riemannian distance on the Stiefel manifold $\mathrm{St}(n,p)$, the set of $n \times p$ matrices with orthonormal columns. The proposed method is a shooting method in the sense of the classical shooting methods for solving boundary value problems; see, e.g., Stoer and Bulirsch, 1993. The main feature is that we provide an approximate formula for the Fréchet derivative of the geodesic involved in our shooting method. Numerical experiments demonstrate the algorithm's accuracy and performance. Comparisons with existing state-of-the-art algorithms for solving the same problem show that our method is competitive and even beats several algorithms in many cases.

Key words. Stiefel manifold, shooting methods, endpoint geodesic problem, Riemannian distance, Newton's method, Fréchet derivative

AMS subject classifications. 65L10, 65F45, 65F60, 65L05, 53C22, 58C15

1. Introduction.

The object of study in this paper is the compact Stiefel manifold, i.e., the set of $n \times p$ matrices with orthonormal columns

$$\mathrm{St}(n,p) = \left\{ X \in \mathbb{R}^{n\times p} : \; X^\top X = I_p \right\}.$$

There are applications in several areas of mathematics and engineering that deal with data that belong to $\mathrm{St}(n,p)$. Domains of applications include numerical optimization, imaging, and signal processing. Some applications, like finding the Riemannian center of mass, require evaluating the geodesic distance between two arbitrary points on $\mathrm{St}(n,p)$. Since no explicit formula is known for computing the distance on $\mathrm{St}(n,p)$, one has to resort to numerical methods.

In this paper we are concerned with computing the Riemannian distance between two given points on the Stiefel manifold. As we shall see, the distance between two points on a manifold is related to the concept of minimizing geodesics[1]. The problem can be briefly formulated as follows. Given two points X, Y on $\mathrm{St}(n,p)$ that are sufficiently close to each other, finding the distance between them is equivalent to finding the tangent vector $\xi^* \in \mathrm{T}_X\mathrm{St}(n,p)$ with the shortest possible length such that [4, 11]

$$\mathrm{Exp}_X(\xi^*) = Y,$$

where Exp_X denotes the Riemannian exponential mapping at X. The solution to this problem is equivalent to the Riemannian logarithm of Y with base point X

$$\xi^* = \mathrm{Log}_X(Y).$$

The sought distance between X and Y is then given by the norm of ξ^*.

Figure 1.1 provides an artistic illustration of the problem. The latter will be stated in more detail in Section 3.

*Received April 5, 2024. Accepted June 26, 2024. Published online on September 11, 2024. Recommended by Peter Benner.

†Mathematics Division, National Center for Theoretical Sciences, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan (msutti@ncts.ntu.edu.tw). ORCID: 0000-0002-8410-1372.

[1]Geodesics are generally defined as critical points of the length functional, and, as such, they may or may not be minima. A minimizing geodesic is one that minimizes the length functional. We introduce the notion of geodesics in Section 2.2.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

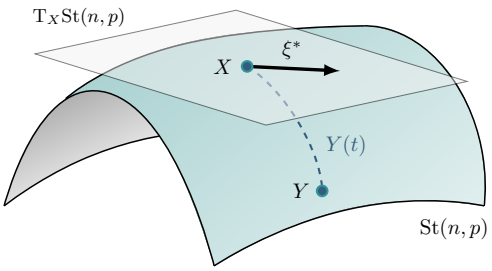

FIG. 1.1. *Illustration of the problem statement.*

It is interesting to note that, for some manifolds, explicit formulas exist for computing the Riemannian distance. This is the case of the Grassmann manifold $\mathrm{Grass}(n,p)$, which is the set of p-dimensional vector subspaces of $\mathbb{R}^n$. For instance, let $\mathcal{X}$ and $\mathcal{Y}$ be two subspaces belonging to $\mathrm{Grass}(n,p)$. Then the distance between $\mathcal{X}$ and $\mathcal{Y}$ is

$$d(\mathcal{X},\mathcal{Y}) = \sqrt{\theta_1^2 + \cdots + \theta_p^2},$$

where θ_i, $i = 1,\ldots,p$, are the principal angles between $\mathcal{X}$ and $\mathcal{Y}$; see, e.g., [27, Theorem 8] and [1, Section 3.8]. The unit sphere $\mathcal{S}^{n-1}$ embedded in $\mathbb{R}^n$ also has explicit formulas for computing the Riemannian distance. In contrast, no such closed-form solution is known for the Stiefel manifold. This motivates us to consider numerical methods. In general, the problem of finding the distance given two points on a Riemannian manifold is related to the Riemannian logarithm function (more details later in Section 2.2). Several authors have already tackled the problem of computing the Riemannian logarithm on the Stiefel manifold. These contributions are detailed in Section 1.3.

1.1. Contributions. In this work, we use the shooting method, which is a classical numerical algorithm for solving boundary value problems, to compute the distance on the Stiefel manifold $\mathrm{St}(n,p)$. These methods are not new (thorough coverage of the shooting methods is given, e.g., in [20]), but their application to computing the Riemannian distance on the Stiefel manifold is relatively new. The method of Bryner, although also named "shooting" in [5], cannot be regarded as a classical shooting method since it makes use of Riemannian geometry concepts (like the parallel transport) that do not fit into the classical framework of [20].

In an earlier version of this work [22, 23], we used the vectorization operator and Kronecker products to work out explicit expressions for the Jacobian matrices involved in the shooting method. The reason why this was done was to carry out some preliminary analysis on the explicit expressions of the Jacobian matrices. The drawback was the excessive computational cost given the dimensions of the operators involved, even for small values of n and p.

Here, in contrast, we work directly with matrices, and we use an approximate form of the Fréchet derivative of the geodesic given by a truncated Fréchet derivative of the matrix exponential. Hence, there is no need for finite difference approximations. In particular, the main contributions of this paper are as follows:

(i) We provide a single shooting method for computing geodesics on the Stiefel manifold using the canonical metric as a classical numerical algorithm for solving boundary value problems.

(ii) We introduce a truncated Fréchet derivative that leads to a linear matrix equation that can be efficiently solved to find the algorithmic update.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

(iii) We perform extensive numerical experiments to demonstrate the algorithm in terms of performance and accuracy. In particular, comparisons with existing state-of-the-art algorithms for solving the same problem show that our method is competitive and even beats several algorithms in many cases.

1.2. Applications and motivation. Many scientific and engineering works have used the Stiefel manifold in their applications. To provide some motivation for the present work, this section summarizes a few applications that explicitly compute the geodesic distance.

In affine invariant shape analysis, Younes et al. [29] studied a specific metric on plane curves that has the property of being isometric to classical manifolds (like the sphere, complex projective plane, Stiefel and Grassmann manifolds) modulo a change of parametrization. Moreover, they provided experimental results that explicitly compute minimizing geodesics between two closed curves.

In the context of shape analysis of closed curves, Srivastava and Klassen [19] studied the space of functions representing unit-length, planar, closed curves, which can be shown to be a Stiefel manifold. Ring and Wirth [18, Section 4.2] provided an application for image segmentation on the Stiefel manifold using a Riemannian variant of the classical BFGS algorithm. This is compared to the work of Sundaramoorthi et al. [21], where the authors used geodesic retractions based on the matrix exponential. The more general reference by Kendall et al. [10, Chapter 6] also contains a discussion on the Stiefel manifold and shape spaces. Bryner [5] also proposed some numerical applications on the pre-shape space.

Çetingül and Vidal [6] investigated the intrinsic mean shift algorithm for clustering on Stiefel and Grassmann manifolds. Turaga et al. [25, 26] investigated applications of the Stiefel manifold in computer vision and pattern recognition to develop accurate inference algorithms. Vision applications such as activity recognition, video-based face recognition, shape classification, and unsupervised clustering were targeted. In particular, step 3 of Algorithm 1 in [26] computes the inverse exponential map, but it was unclear how this was achieved.

The low-rank representation (LRR) is a widely used technique in computer vision and pattern recognition for data clustering models. Yin et al. [28] extended the LRR from Euclidean space to manifold-valued data on the Stiefel manifold by incorporating the intrinsic geometry of the manifold. They acknowledged that, in general, it is pretty hard to compute the log mapping for the Stiefel manifold. Consequently, they used the retraction map (a first-order approximation of the exponential mapping; see, e.g., [3]) instead of the exponential map because of its reduced computational cost.

More recently, Li and Ma [12] proposed a generalization of the federated learning framework to Riemannian manifolds. In particular, they considered the kPCA problem on the Stiefel manifold. Even though they initially discussed the Riemannian logarithm mapping, they finally adopted a retraction in the numerical implementations, similarly to what was done in [28].

1.3. Related works and other approaches. Shooting methods are not the only option to solve the endpoint geodesic problem; many other numerical algorithms have been proposed. As a plethora of methods now come out every year, this review is not meant to be exhaustive.

The leapfrog algorithm by Noakes [14] is based on partitioning the original problem into smaller subproblems. This method has global convergence properties, but it slows down for an increasing number of subproblems or when the solution is approached [9, Section 1]. Moreover, Noakes realized that his leapfrog algorithm was in some way imitating the Gauss–Seidel method [14, Section 1]. This connection has been explored by Sutti and Vandereycken [24].

Bryner [5] proposed two schemes, named “shooting method” and “path-straightening”, to compute endpoint geodesics on the Stiefel manifold by considering it as an embedded

ETNA
Kent State University and
Johann Radon Institute (RICAM)

submanifold of the Euclidean space. From the matrix algebra perspective, Rentmeesters [17] and Zimmermann [30, 31] derived algorithms for evaluating the Riemannian logarithm map on the Stiefel manifold with respect to the canonical metric, which are locally convergent and depend upon the definition of the matrix logarithm function. Recently, Zimmermann and Hüper [32] provided a unified method to deal with the endpoint geodesic problem on the Stiefel manifold with respect to a family of metrics.

Noakes and Zhang [15] proposed an alternative algorithm to find geodesics joining two given points. Like leapfrog, this method exploits the shooting method to compute geodesics joining junction points.

The methods of Nguyen [13] are based on classical (Euclidean) optimization algorithms for minimizing an objective function with a reduction of the computational cost thanks to the formulation of the gradients using only the Fréchet derivatives.

1.4. Notation. In Table A.1 (see Appendix A) we list the notations and symbols adopted in this paper in order of appearance. Symbols only used in one section are typically omitted from this list.

1.5. Outline of the paper. The remaining part of this paper is organized as follows. Section 2 introduces the geometry of the Stiefel manifold. Readers who are familiar with Riemannian geometry, particularly the geometry of the Stiefel manifold, might want to skip this section. Section 3 presents the problem statement, which is the focus of this work. Section 4 describes our proposed algorithm: a single shooting method with an approximate Fréchet derivative. Numerical experiments and comparisons with other methods are presented in Section 5. Finally, we conclude the paper by summarizing the contributions and providing future research outlooks in Section 6.

2. Geometry of the Stiefel manifold. This section introduces the geometry of the Stiefel manifold. Here, we only give the necessary background to understand the remaining part of this paper. For additional details, we refer the reader to the reference works of [2, 4, 7].

The set of all $n \times p$ orthonormal matrices, i.e.,

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n\times p} \colon X^\top X = I_p\},$$

endowed with its submanifold structure is called an orthogonal or compact Stiefel manifold. It is a subset of $\mathbb{R}^{n\times p}$, and it can be proven that it has the structure of an embedded submanifold of $\mathbb{R}^{n\times p}$ [2, Section 3.3.2]. The Stiefel manifold $\mathrm{St}(n,p)$ may also degenerate to some special cases. For $p = 1$, it reduces to the unit sphere $\mathcal{S}^{n-1}$ in $\mathbb{R}^n$, while for $p = n$, it becomes the orthogonal group $\mathrm{O}(n)$, whose dimension is $\frac{1}{2}n(n-1)$.

2.1. Tangent spaces and projectors. The tangent space to a manifold at a given point can be seen as a local vector space approximation of the manifold at that point. In practice, the tangent space is used to perform the operations of vector addition and scalar multiplication, which would otherwise be impossible to perform on the manifold without leaving it due to the manifold's curvature. Endowed with a Euclidean inner product, this vector space becomes a Euclidean space where we also have a notion of length. Here, we will directly focus on the tangent space to the Stiefel manifold. For a more precise definition of a tangent space in the general case, we refer the reader to [2].

The tangent space to the Stiefel manifold at a point X can be characterized by [2, Example 3.5.2]

$$\mathrm{T}_X\mathrm{St}(n,p) = \{X\Omega + X_\perp K \colon \Omega = -\Omega^\top,\ K \in \mathbb{R}^{(n-p)\times p}\}, \tag{2.1}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Ω being a p-by-p skew-symmetric matrix, $\Omega \in \mathcal{S}_{\text{skew}}(p)$, $X_\perp$ being an orthonormal matrix whose columns span the orthogonal complement of $\operatorname{span}(X)$, and $K \in \mathbb{R}^{(n-p)\times p}$, with no restriction on K. With this characterization in mind, and with the fact that $\dim\big(\operatorname{St}(n,p)\big) = \dim\big(\mathrm{T}_X\operatorname{St}(n,p)\big)$, it is straightforward to calculate the dimension of the Stiefel manifold as

$$\dim(\operatorname{St}(n,p)) = \dim(\mathcal{S}_{\text{skew}}(p)) + \dim(\mathbb{R}^{(n-p)\times p}) = \tfrac{1}{2}p(p-1) + (n-p)p = np - \tfrac{1}{2}p(p+1).$$

The projection onto the tangent space $\mathrm{T}_X\operatorname{St}(n,p)$ is

$$\mathrm{P}_X\,\xi = X\operatorname{skew}(X^\top\xi) + (I - XX^\top)\,\xi. \tag{2.2}$$

2.2. Geodesics, exponential mapping, and logarithm mapping. Geodesics are defined in general as curves with zero covariant acceleration. They allow us to introduce the *Riemannian exponential* $\operatorname{Exp}_x\colon \mathrm{T}_x\mathcal{M} \to \mathcal{M}$ that maps a tangent vector $\xi = \dot\gamma(0) \in \mathrm{T}_x\mathcal{M}$ to the geodesic endpoint $\gamma(1) = y$: $\operatorname{Exp}_x(\xi) = y$. Figure 2.1 illustrates these concepts for the unit sphere $\mathcal{S}^2$, which is also a special case of a Stiefel manifold $\operatorname{St}(n,p)$, with $n = 3$ and $p = 1$.

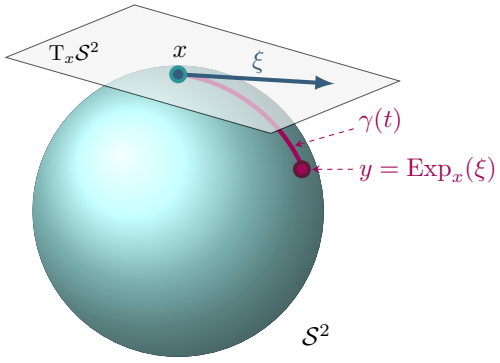

FIG. 2.1. *The Riemannian exponential map on the sphere.*

To define a distance on a given manifold $\mathcal{M}$, we need a notion of length that applies to tangent vectors. To this aim, we endow the tangent space $\mathrm{T}_x\mathcal{M}$ with an *inner product* $\langle\cdot,\cdot\rangle_x$. The inner product $\langle\cdot,\cdot\rangle_x$ induces a norm $\|\xi_x\|_x = \sqrt{\langle\xi_x,\xi_x\rangle_x}$ on $\mathrm{T}_x\mathcal{M}$. A manifold $\mathcal{M}$ endowed with a smoothly varying inner product (called *Riemannian metric* g) is called *Riemannian manifold*.

The *length of a curve* $\gamma\colon [a,b] \to \mathcal{M}$ on a Riemannian manifold $(\mathcal{M}, g)$ is

$$L(\gamma) = \int_a^b \sqrt{g(\dot\gamma(t),\dot\gamma(t))}\,\mathrm{d}t.$$

The *Riemannian distance* is defined as the shortest path between two points x and y

$$d\colon \mathcal{M}\times\mathcal{M} \to \mathbb{R}\colon d(x,y) = \inf_{\gamma\in\Gamma} L(\gamma),$$

where Γ denotes the set of all curves γ in $\mathcal{M}$ joining the points x and y.

Generally speaking, different choices of a Riemannian metric are possible. In this paper, we consider the non-Euclidean *canonical metric* inherited by $\operatorname{St}(n,p)$ from its definition as a

ETNA
Kent State University and
Johann Radon Institute (RICAM)

quotient space of the orthogonal group [7, (2.39)]. Given $X \in \mathrm{St}(n,p)$ and $\xi, \zeta \in \mathrm{T}_X\mathrm{St}(n,p)$, the canonical metric reads

$$g_c(\xi,\zeta) = \mathrm{Tr}\big(\xi^\top (I - \tfrac{1}{2} X X^\top)\, \zeta\big).$$

The canonical metric induces the *canonical norm*, defined as

$$\|\xi\|_{\mathrm{c}} = \sqrt{g_c(\xi,\xi)}.$$

The reader can verify that

$$\|\xi\|_{\mathrm{c}}^2 = \tfrac{1}{2}\|\Omega\|_{\mathrm{F}}^2 + \|K\|_{\mathrm{F}}^2.$$

The *embedded metric* is the metric inherited by the Stiefel manifold as an embedded submanifold of $\mathbb{R}^n$, i.e., $g_e(\xi,\xi) = \mathrm{Tr}(\xi^\top \xi)$. With the embedded metric, the induced norm is simply the Frobenius norm

$$\|\xi\|_{\mathrm{e}} = \sqrt{g_e(\xi,\xi)} = \sqrt{\mathrm{Tr}(\xi^\top \xi)} =: \|\xi\|_{\mathrm{F}} \qquad \text{and} \qquad \|\xi\|_{\mathrm{F}}^2 = \|\Omega\|_{\mathrm{F}}^2 + \|K\|_{\mathrm{F}}^2.$$

The only difference with respect to the squared canonical norm of ξ is that in the squared embedded norm of ξ the term $\|\Omega\|_{\mathrm{F}}^2$ is not halved. This calculation highlights the fact that in contrast to the embedded norm, the canonical norm only takes into account once the $\frac{1}{2}p(p-1)$ coefficients of Ω. Indeed, in the remaining part of this paper, we will only use the canonical metric.

By endowing the Stiefel manifold with the canonical metric, one can derive the following *second-order ordinary differential equation for the geodesic* [7, (2.41)]

$$\ddot{Y} + \dot{Y}\dot{Y}^\top Y + Y\big((Y^\top \dot{Y})^2 + \dot{Y}^\top \dot{Y}\big) = 0, \tag{2.3}$$

where $Y \equiv Y(t)$. An explicit formula for a geodesic that realizes a tangent vector ξ with base point Y_0 is [7, (2.42)]

$$Y(t) = Q \exp_{\mathrm{m}}\left(\begin{bmatrix} \Omega & -K^\top \\ K & O_{n-p} \end{bmatrix} t\right) \cdot I_{n,p}, \tag{2.4}$$

with $Q = \begin{bmatrix} Y_0 & Y_{0\perp} \end{bmatrix}$ and $Y_{0\perp}$ being any matrix whose columns span $\mathcal{Y}_0^\perp = (\mathrm{span}(Y_0))^\perp$. If $t = 1$, then this is precisely the Riemannian exponential on the Stiefel manifold. In this paper, we denote by A the matrix in the argument of the matrix exponential $\exp_{\mathrm{m}}$.

REMARK 2.1. The matrix $Y_{0\perp}$ does not need to be orthonormal. Indeed, its only requirement is that it has to span $\mathcal{Y}_0^\perp = (\mathrm{span}(Y_0))^\perp$, i.e., the orthogonal subspace to $\mathcal{Y}_0 = \mathrm{span}(Y_0)$; see [22, Appendix A.1]. For the convenience of our analysis and implementation, we always assume that $Y_{0\perp}$ is orthonormal so that $Q = \begin{bmatrix} Y_0 & Y_{0\perp} \end{bmatrix}$ is an orthogonal matrix.

REMARK 2.2. It can be shown that the endpoint geodesic problem on $\mathrm{St}(n,p)$ is equivalent to an endpoint geodesic problem on $\mathrm{St}(2p,p)$; see [7, 17] and [22, Section 2.3.3 and Appendix A.2]. In the formulation (2.4) above, the complexity of computing the matrix exponential is $O(n^3)$, but if $p \ll n$, then the smaller formulation can be used and its computational cost is only $O(p^3)$. In practice, it makes sense to consider the formulation on $\mathrm{St}(2p,p)$ only if $p < \frac{n}{2}$, which is what we do in our algorithm.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

3. Problem statement. In this section, we state the problem more formally. Given two points Y_0, Y_1 on $\mathrm{St}(n,p)$ that are sufficiently close to each other, finding the distance between them is equivalent to finding the tangent vector $\xi^* \in \mathrm{T}_{Y_0}\mathrm{St}(n,p)$ with the shortest possible length such that [4, 11]

$$\mathrm{Exp}_{Y_0}(\xi^*) = Y_1,$$

where Exp_{Y_0} denotes the Riemannian exponential mapping at Y_0. The solution to this problem is equivalent to the Riemannian logarithm of Y_1 with base point Y_0

$$\xi^* = \mathrm{Log}_{Y_0}(Y_1).$$

We refer the reader to Figure 1.1 for an illustration of the problem statement.

In terms of the differential equation (2.3) governing the geodesic, the problem statement may be written as follows:

Find $\xi^* \equiv \dot{Y}(0) \in \mathrm{T}_{Y_0}\mathrm{St}(n,p)$ such that the second-order ODE

$$\ddot{Y} = -\dot{Y}\dot{Y}^\top Y - Y\big((Y^\top\dot{Y})^2 + \dot{Y}^\top\dot{Y}\big) \quad \text{with boundary conditions} \quad \begin{cases} Y(0) = Y_0, \\ Y(1) = Y_1, \end{cases} \tag{3.1}$$

is satisfied. This problem is known as a *boundary value problem* (BVP).

4. A single shooting method with an approximate Fréchet derivative. The single shooting method is a classical numerical scheme for solving boundary value problems. The main idea is to reformulate the BVP as an initial value problem (IVP), guess the initial value of the acceleration, and then solve a nonlinear equation. It basically turns a BVP into a root-finding problem. The zeros of the nonlinear equation can be computed with any root-finding algorithm, but the classical single shooting method typically uses Newton's method.

In this section we present the details on how to apply the single shooting method to the endpoint geodesic problem on the Stiefel manifold. We start by recasting the BVP (3.1) into an IVP. Let $Z_1(t) = Y(t)$, $Z_2(t) = \dot{Y}(t)$ denote the geodesic and its derivative, respectively, and let

$$Z(t) = \begin{bmatrix} Z_1(t) \\ Z_2(t) \end{bmatrix}.$$

We get the initial value problem (we omit the dependence on t)

$$\dot{Z} = \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \end{bmatrix} = \begin{bmatrix} Z_2 \\ -Z_2Z_2^\top Z_1 - Z_1\big((Z_1^\top Z_2)^2 + Z_2^\top Z_2\big) \end{bmatrix}, \quad \text{with initial conditions } Z(0) = \begin{bmatrix} Z_1(0) \\ Z_2(0) \end{bmatrix} = \begin{bmatrix} Y_0 \\ \xi \end{bmatrix}. \tag{4.1}$$

Here, ξ is the unknown such that $Z_1(1) = Y_1$.

Solving (4.1) typically requires a numerical integration scheme, but here, since we already have the explicit formula (2.4) for the geodesic $Z_1(t)$, we do not need to integrate the initial value problem (4.1).

The explicit formula for Z_2 is just the derivative of Z_1 with respect to t, namely,

$$Z_2(t) = Q\,\exp_{\mathrm{m}}\left(\begin{bmatrix} \Omega & -K^\top \\ K & O_{n-p} \end{bmatrix} t\right)\begin{bmatrix} \Omega \\ K \end{bmatrix},$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $\Omega = Y_0^\top \xi$ and $K = Y_{0\perp}^\top \xi$. Now, let us define the function

$$F(\xi) = Z_1(1, \xi) - Y_1, \tag{4.2}$$

where we emphasize the dependence on ξ. Roughly speaking, this represents the mismatch between $Z_1(1, \xi)$, i.e., the geodesic at $t = 1$, and the boundary condition Y_1 we wish to enforce. Our goal is to find ξ^* such that

$$F(\xi^*) = 0.$$

As mentioned above, this is a root-finding problem of a nonlinear (matrix) equation, which can be solved by *Newton's method*. To apply Newton's method, we need the Fréchet derivative of $Z_1(1, \xi)$ in the direction of an increment $\delta\xi$ of ξ.

In an earlier version of this work [22, 23], we used a vectorization approach that allowed us to calculate an explicit analytic expression for the Jacobian matrix involved in this single shooting method. The drawback of this approach is that it is computationally very inefficient due to the dimensions of the operators involved, which grow exponentially with n. Here, instead, we work directly with matrix equations and Fréchet derivatives. In doing so, we drew some inspiration from [13].

In the remaining part of this section, we first state the algorithm and then, in Section 4.1, explain in more detail the linearization of (4.2) and the approximation of the Fréchet derivative involved. In Section 4.2, we provide a way to construct an initial iterate for the algorithm.

The pseudocode for the single shooting method on the Stiefel manifold is given in Algorithm 1. Note at line 8 the additional projection step onto the tangent space to ensure that the updated tangent vector is indeed an element of $\mathrm{T}_{Y_0}\mathrm{St}(n, p)$. We did not take any particular care in introducing a line-search technique, although it might be helpful for globalizing the method. Numerical experiments in Section 5 demonstrate that Algorithm 1 works very well in practice. For the sake of brevity, we name our algorithm SSAF (an acronym for "single shooting approximate Fréchet"). As a stopping criterion, we typically consider a tolerance for the norm of the residual $\delta\xi^{(k)}$.

Algorithm 1: A single shooting method on the Stiefel manifold with an approximation of the Fréchet derivative (SSAF method).

1 Given Y_0, Y_1;
  Result: ξ^* such that $\mathrm{Exp}_{Y_0}(\xi^*) = Y_1$.
2 Compute the initial guess $\xi^{(0)}$ (using Algorithm 2);
3 Set $k = 0$;
4 **while** *a stopping criterion is met* **do**
5 Compute $F^{(k)} = Z_1(1, \xi^{(k)}) - Y_1$;
6 Solve $F^{(k)} + \mathrm{D}\, Z_1\left[\delta\xi^{(k)}\right] = 0$ for $\delta\xi^{(k)}$;
7 Update $\xi^{(k+1)} \leftarrow \xi^{(k)} + \delta\xi^{(k)}$;
8 Project $\xi^{(k+1)}$ onto $\mathrm{T}_{Y_0}\mathrm{St}(n, p)$ using (2.2): $\xi^{(k+1)} \leftarrow \mathrm{P}_{Y_0}\left(\xi^{(k+1)}\right)$;
9 $k = k + 1$;
10 **end while**

4.1. Linearization of the nonlinear matrix equation (4.2). We recall from (2.1) the structure of a tangent vector $\xi \in \mathrm{T}_{Y_0}\mathrm{St}(n, p)$, i.e.,

$$\xi = Y_0\Omega + Y_{0\perp}K.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

From now on, let us denote by

$$A(\xi) = \begin{bmatrix} \Omega & -K^\top \\ K & O_{n-p} \end{bmatrix}$$

the matrix in the argument of the exponential appearing in the geodesic equation (2.4). Clearly, A is a function of ξ because the matrices Ω and K are formed from the tangent vector ξ. Then (2.4) at $t = 1$ can be rewritten as

$$Z_1(1, \xi) = Q \exp_{\mathrm{m}}(A(\xi)) \cdot I_{n,p}.$$

Recall our nonlinear matrix equation (4.2) that we want to solve for ξ. Newton's method consists in solving successive linearizations of (4.2), i.e.,

$$F(\xi + \delta\xi) = Z_1(\xi + \delta\xi) - Y_1 = 0. \tag{4.3}$$

Here, the term $Z_1(\xi + \delta\xi)$ is the expression for the geodesic when applying a small perturbation $\delta\xi$ to the vector ξ. Applying matrix perturbation theory, we obtain

$$Z_1(\xi + \delta\xi) = Z_1(\xi) + \mathrm{D}\, Z_1\, [\delta\xi^{(k)}] + o(\|\delta\xi\|),$$

namely,

$$Z_1(\xi + \delta\xi) = Z_1(\xi) + Q\ \mathrm{D}\exp_{\mathrm{m}}(A(\xi))\big[\,\mathrm{D}A(\xi)[\delta\xi]\big] \cdot I_{n,p} + o(\|\delta\xi\|), \tag{4.4}$$

where $\mathrm{D}\exp_{\mathrm{m}}(A(\xi))\big[\,\mathrm{D}A(\xi)[\delta\xi]\big]$ denotes the Fréchet derivative of the matrix exponential at $A(\xi)$ in the direction of $\mathrm{D}A(\xi)[\delta\xi]$. $\mathrm{D}A(\xi)[\delta\xi]$ itself denotes the Fréchet derivative of $A(\xi)$ in the direction of $\delta\xi$.

In contrast to what was proposed in [22, 23], here we do not vectorize the equation, and we work directly with matrices and Fréchet derivatives; moreover, we do not compute the exact Fréchet derivative, but we approximate it by a truncated expansion.

Inserting (4.4) into (4.3) and neglecting the higher-order terms in $\delta\xi$, we obtain the matrix equation

$$Z_1(\xi) + Q\ \mathrm{D}\exp_{\mathrm{m}}(A(\xi))\big[\,\mathrm{D}A(\xi)[\delta\xi]\big] \cdot I_{n,p} - Y_1 = 0. \tag{4.5}$$

We now need to tackle the term $\mathrm{D}\exp_{\mathrm{m}}(A(\xi))\big[\,\mathrm{D}A(\xi)[\delta\xi]\big]$, which involves a chain rule with two Fréchet derivatives.

First, the perturbation of $A(\xi)$ with a $\delta\xi$ gives

$$A(\xi + \delta\xi) = A(\xi) + \mathrm{D}A(\xi)[\delta\xi],$$

where

$$A(\xi) = \begin{bmatrix} \Omega & -K^\top \\ K & O_{(n-p)} \end{bmatrix} \qquad \text{and} \qquad \mathrm{D}A(\xi)[\delta\xi] = \begin{bmatrix} \delta\Omega & -\delta K^\top \\ \delta K & O_{(n-p)} \end{bmatrix}.$$

Since A is linear in ξ, the above expansion is exact.

Secondly, the perturbation of the matrix exponential by a matrix $E \in \mathbb{R}^{n\times n}$ is

$$\exp_{\mathrm{m}}(A + E) = \exp_{\mathrm{m}}(A) + \mathrm{D}\exp_{\mathrm{m}}(A)[E] + o(\|E\|),$$

where $\mathrm{D}\exp_{\mathrm{m}}(A)[E]$ is the Fréchet derivative of the matrix exponential at A in the direction of E. In general, there are many ways to compute $\exp_{\mathrm{m}}$ and $\mathrm{D}\exp_{\mathrm{m}}(A)[E]$ and thus also

ETNA
Kent State University and
Johann Radon Institute (RICAM)

many ways to approximate these quantities; see [8, Chapter 10]. Here, we consider the Taylor series of $e^{A+E} - e^A$, from which we obtain the following representation for the Fréchet derivative of the matrix exponential [8, Section 10.2]

$$\mathrm{D}\exp_{\mathrm{m}}(A)[E] = E + \frac{AE+EA}{2} + \frac{A^2E + AEA + EA^2}{3!} + \cdots .$$

We then consider an approximation of $\mathrm{D}\exp_{\mathrm{m}}(A)[E]$ by keeping only the first two terms in the expansion, i.e.,

$$\mathrm{D}\exp_{\mathrm{m}}(A)[E] \approx E + \frac{AE+EA}{2}.$$

This formula can be used to approximate $\mathrm{D}\exp_{\mathrm{m}}(A(\xi))\big[\mathrm{D}A(\xi)[\delta\xi]\big]$ in (4.5), resulting in

$$Q \cdot \big(\mathrm{D}A(\xi)[\delta\xi] + \tfrac{1}{2}(A \cdot \mathrm{D}A(\xi)[\delta\xi] + \mathrm{D}A(\xi)[\delta\xi] \cdot A)\big) \cdot I_{n,p} = Y_1 - Z_1.$$

This is now a linear matrix equation to be solved for $\delta\xi$. In practice, we work with the factors $\delta\Omega$ and δK of $\delta\xi$. After a few algebraic manipulations, detailed in Appendix B, we obtain a (small-sized) Sylvester equation which can be efficiently solved with MATLAB's command `lyap` to obtain the update $\delta\Omega$. Then the update δK can be found from $\delta\Omega$. Let the current iteration be indexed by k. Then the tangent vector is updated as

$$\xi^{(k+1)} = \xi^{(k)} + Q \cdot \begin{bmatrix} \delta\Omega^{(k)} \\ \delta K^{(k)} \end{bmatrix}.$$

4.2. The initial guess. This section outlines our approach to initializing our single shooting method, which involves choosing an initial guess $\xi^{(0)}$ that is close enough to ξ^*. It is well known that Newton's method exhibits only local convergence properties, which means that the method requires a sufficiently good initial guess to converge. Shortcomings of Newton's method are very well described in [16, Section 11.1]. It is possible to modify Newton's method and enhance it in various ways to get around most of these problems.

As Newton's method underlies the single shooting method, selecting a "good enough" initial guess is crucial. Although in this work we actually consider an approximation of the Fréchet derivative and not the exact one, we can still use the following construction of the initial guess that was used in [22, 23]. This construction is closely related to the "first shot" in Bryner's method [5, Alg. 1].

Concretely, we use a first-order approximation of the matrix exponential $\exp_{\mathrm{m}}(A) \approx I + A$ appearing in (4.3) and solve for ξ. This yields a first-order approximation $\bar{\xi}$ of the solution ξ^* as

$$\bar{\xi} = Y_1 - Y_0.$$

Since, in general, this is no longer an element of the tangent space, we need to project it onto $\mathrm{T}_{Y_0}\mathrm{St}(n,p)$ to obtain a tangent vector. We expect the tangent vector so obtained to be a satisfactory initial approximation of the sought tangent vector ξ^*.

Using (2.2), the projection of $\bar{\xi}$ onto the tangent space at Y_0 is

$$\mathrm{P}_{Y_0}\bar{\xi} = Y_0\,\mathrm{skew}\big(Y_0^\top(Y_1 - Y_0)\big) + (I_n - Y_0Y_0^\top)(Y_1 - Y_0) = Y_1 - Y_0\,\mathrm{sym}(Y_0^\top Y_1).$$

To get $\xi^{(0)}$, we rescale this vector so that its norm is equal to the norm of $\bar{\xi}$, i.e.,

$$\xi^{(0)} = \frac{\|\bar{\xi}\|}{\|\mathrm{P}_{Y_0}\bar{\xi}\|}\,\mathrm{P}_{Y_0}\bar{\xi}.$$

This procedure is summarized in Algorithm 2 and illustrated in Figure 4.1.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

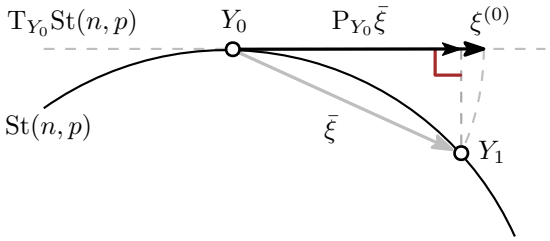

FIG. 4.1. *Initial guess for the single shooting method on the Stiefel manifold.*

Algorithm 2: Initial guess for the single shooting method on the Stiefel manifold.

1 Given Y_0, Y_1;
2 Compute $\bar{\xi} = Y_1 - Y_0$;
3 Compute $\mathrm{P}_{Y_0}\bar{\xi} = Y_1 - Y_0\,\mathrm{sym}(Y_0^\top Y_1)$;
4 Compute $\xi^{(0)} = \dfrac{\|\bar{\xi}\|}{\|\mathrm{P}_{Y_0}\bar{\xi}\|}\,\mathrm{P}_{Y_0}\bar{\xi}$;
5 Return $\xi^{(0)}$.

5. Numerical experiments and comparisons with other methods. In this section, we present some numerical experiments for the single shooting method, and we report on the convergence behavior. The code was implemented in MATLAB and is freely available on the repository https://github.com/MarcoSutti/SSAF_2024_repo. The method of Bryner [5, Alg. 1] was implemented according to the pseudocode provided in [5], while the method of Zimmermann [5, Alg. 1] was directly taken from Appendix C of [30]. We conducted our experiments on a laptop Lenovo ThinkPad T460s with Ubuntu 23.10 LTS and MATLAB R2022a installed, with Intel Core i7-6600 CPU, 20GB RAM, and Mesa Intel HD Graphics 520. On this machine, the matrix exponential `expm` of a unit-norm skew-symmetric matrix in $\mathbb{R}^{1000\times 1000}$ is computed in 0.45 seconds (time averaged over 100 runs)[2]. In all the tables, our algorithm is named SSAF.

Table 5.1 compares our new SSAF method with the earlier version using the exact Jacobian matrix [22, Alg. 1]. The efficiency of our new SSAF method compared to the old version is striking. Typically, cases with a small p require more iterations for SSAF, while for [22, Alg. 1] the number of iterations remains constant with respect to p. Still, the superior efficiency of SSAF by far compensates for this (desirable) feature. From Table 5.1, it is also evident that the single shooting method with the exact Jacobian matrix [22, Alg. 1] scales very badly with p, and it becomes prohibitively expensive as p grows. The long dashes "—" in the table indicate that the single shooting method with the exact Jacobian matrix stopped due to memory overflow.

5.1. Comparisons with other state-of-the-art methods. In this section we demonstrate that the proposed algorithm is competitive with other state-of-the-art methods. The body of literature and available methods has been increasing recently, especially during the last five years, and it would be hard to compare our proposed algorithm to all the existing methods. Hence, the comparisons in this section are not meant to be exhaustive. We aim to show that our proposed algorithm is competitive with respect to only a limited subset of state-of-the-art algorithms that can be found in the literature. Specifically, in this section we compare our

[2]We give this reference for the computational cost of the matrix exponential `expm` following [13, Section 5.2].

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 5.1
Comparisons for the single shooting method with an approximate Fréchet derivative versus the single shooting method with the exact Jacobian [22, Alg. 1] on $\mathrm{St}(1000, p)$ *for doubling values of* p *and for a prescribed* $d(X, Y) = 0.5\,\pi$. *Results are averaged over 10 random runs. The stopping tolerance is* 10^{-5}.

p	Avg. comput. time (s)		Avg. no. of iterations	
	[22, Alg. 1]	SSAF	[22, Alg. 1]	SSAF
10	0.08	0.00256	5	6.80
20	3.07	0.00390	5	5.00
40	182.69	0.00998	5	5.00
80	—	0.03388	—	4.00
160	—	0.12734	—	4.00
320	—	0.90012	—	4.00
640	—	2.98705	—	4.00

method to the "shooting" method of Bryner [5, Alg. 1], the matrix algebraic approach of Zimmermann [30, Alg. 1], and the optimization methods of Nguyen [13].

Table 5.2 uses the same test cases as those considered in [13, Table 2], namely $\mathrm{St}(1500, p)$ for large values of p and for a prescribed $d(X, Y) = 0.5\,\pi$[3]. In all the test cases considered, our SSAF algorithm is superior in terms of computation time compared to the other methods.

TABLE 5.2
Comparisons on $\mathrm{St}(1500, p)$ *with large values of* p *and for a prescribed* $d(X, Y) = 0.5\,\pi$. *Results are averaged over 10 pairs of randomly generated endpoints on* $\mathrm{St}(1500, p)$. *The stopping tolerance is* 10^{-5}.

p	Avg. comput. time (s)			Avg. no. of iterations		
	[5, Alg. 1]	[30, Alg. 1]	SSAF	[5, Alg. 1]	[30, Alg. 1]	SSAF
500	12.09	3.30	1.89	3.00	2.00	4.00
700	31.27	8.21	4.56	3.00	2.00	4.00
1000	77.37	20.39	8.82	3.00	2.00	4.00

Here, we do not perform a direct comparison with the methods of [13], in particular with the results in [13, Table 2]. However, we can point out that the results in [13, Table 2] are obtained on a machine that computes the matrix exponential `expm` of a unit-norm skew-symmetric matrix in $\mathbb{R}^{1000\times 1000}$ in 0.6 seconds. In contrast, on our machine, as mentioned at the beginning of this section, this same reference quantity is 0.45 seconds. This suggests that if it were to be run on the same machine, the methods of [13] would be slightly faster but still slower than our SSAF method and also than [30, Alg. 1].

However, when p is doubled from 500 to 1000, the methods of [13, Table 2] seem to scale better than all the other algorithms considered here. We emphasize that the test case corresponding to the last row in Table 5.2 has dimensions $n = 1500$ and $p = 1000$, i.e., this is not a case in which $p \leq n/2$. Therefore, the standard implementations of [5, Alg. 1] and [30, Alg. 1] are not designed to be efficient in this case. From the numerical results reported in [13, Table 2], it seems that when p is doubled from 500 to 1000, the computation time is multiplied by a factor of approximately 3.4 for the gradient descent method and 3.6 for the L-BFGS method. In contrast, the same factors computed from Table 5.2 for [5, Alg. 1], [30, Alg. 1], and our SSAF method are 6.4, 6.2, and 4.7, respectively. This suggests that

[3]This is the prescribed distance used for the numerical experiments reported in [13, Table 2], although not explicitly written in that paper. From a private communication with Nguyen, February 2024.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Nguyen's methods might be more effective for problems with larger values of p. Yet our SSAF method remains the most competitive among the other algorithms considered here because it shows to have a factor of 4.7 in contrast to 6.4 for [5, Alg. 1] and 6.2 for [30, Alg. 1].

Table 5.3 considers the same three test cases as in [32, Table 1] with a stopping tolerance $\tau = 10^{-10}$. Although the average number of iterations for our SSAF method is much higher than that of the other algorithms, our method remains competitive in terms of computation time.

TABLE 5.3
Comparisons for the three test cases in [32, Table 1] with a stopping tolerance $\tau = 10^{-10}$. Results are averaged over 10 experiments.

Avg. comput. time (s)			Avg. no. of iterations		
[5, Alg. 1]	[30, Alg. 1]	SSAF	[5, Alg. 1]	[30, Alg. 1]	SSAF
Test Case 1: $\mathrm{St}(2000, 500)$, for a prescribed $d(X, Y) = 5\pi$.					
109.65	15.35	19.66	14.0	12.0	28.2
Test Case 2: $\mathrm{St}(120, 30)$, for a prescribed $d(X, Y) = \pi$.					
0.11242	0.04157	0.02556	10.90	9.30	19.60
Test Case 3: $\mathrm{St}(12, 3)$, for a prescribed $d(X, Y) = 0.95\pi$.					
0.08885	0.06563	0.03948	40.80	86.30	167.00

Tables 5.4 and 5.5 below try to reproduce the data from the left and middle panels, respectively, of [5, Figure 4], while at the same time comparing with the method of [30] and with our SSAF method. We adopt the same parameters as in [5], namely $T = 20$ and a stopping tolerance $\tau = 10^{-3}$. We say that we "try to reproduce" since it seems that Bryner did not prescribe a distance or at least this is not explicitly stated; hence, we fix it here to $d(X, Y) = 0.5\,\pi$.

TABLE 5.4
Comparisons on $\mathrm{St}(n, 2)$ for doubling values of n and for a prescribed $d(X, Y) = 0.5\,\pi$. $T = 20$ and tolerance $\tau = 10^{-3}$. Results are averaged over 100 experiments.

n	Avg. comput. time (s)			Avg. no. of iterations		
	[5, Alg. 1]	[30, Alg. 1]	SSAF	[5, Alg. 1]	[30, Alg. 1]	SSAF
10	0.00400	0.00091	0.00080	4.08	3.73	7.77
20	0.00367	0.00093	0.00091	3.85	3.87	7.35
40	0.00337	0.00095	0.00075	3.49	3.61	6.96
80	0.00312	0.00101	0.00081	3.30	3.61	6.90
160	0.00310	0.00105	0.00086	3.15	3.42	6.86
320	0.00328	0.00107	0.00096	3.02	3.08	6.86
640	0.00371	0.00105	0.00091	3.00	3.02	6.89
1 280	0.00543	0.00104	0.00100	3.00	2.72	6.87
2 560	0.00856	0.00135	0.00121	3.00	2.47	6.87
5 120	0.01056	0.00131	0.00132	3.00	2.34	6.93
10 240	0.01596	0.00144	0.00141	3.00	2.12	6.97

ETNA
Kent State University and
Johann Radon Institute (RICAM)

From Table 5.5, it appears that the endpoint geodesic problem seems to get easier for large n, since for all the three methods considered, the average number of iterations decreases for increasing n. In other words, as the ratio $p/n \to 0$, solving the endpoint geodesic problem requires fewer iterations. This is consistent with a similar observation in [13, Section 5.2] and with [30, Table 5.1].

TABLE 5.5
Comparisons on $\mathrm{St}(500, p)$ *for doubling values of* p *and for a prescribed* $d(X, Y) = 0.5\,\pi$. $T = 20$ *and tolerance* $\tau = 10^{-3}$. *Results are averaged over 100 experiments.*

p	Avg. comput. time (s)			Avg. no. of iterations		
	[5, Alg. 1]	[30, Alg. 1]	SSAF	[5, Alg. 1]	[30, Alg. 1]	SSAF
2	0.00353	0.00103	0.00086	3.01	2.95	6.78
4	0.00533	0.00156	0.00128	3.00	2.81	5.28
8	0.00711	0.00182	0.00115	3.00	2.00	4.08
16	0.01173	0.00369	0.00173	3.00	2.00	4.00
32	0.02912	0.01354	0.00453	3.00	2.00	4.00
64	0.08762	0.03582	0.01150	3.00	2.00	3.00
128	0.40437	0.10052	0.05657	3.00	1.00	3.00
256	1.94025	0.47720	0.25847	3.00	1.00	3.00

Using the data from Tables 5.4 and 5.5, Figure 5.1 tries to reproduce the left and middle panels of [5, Figure 4], while at the same time comparing with Zimmermann's algorithm [30, Alg. 1] and our SSAF method.

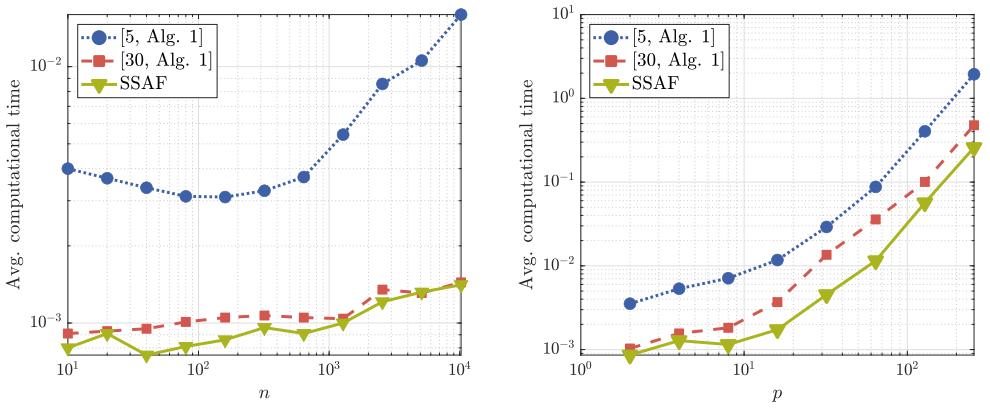

FIG. 5.1. *Average computation times for Bryner's shooting method [5, Alg. 1], Zimmermann's matrix algebraic algorithm [30, Alg. 1], and our SSAF method on* $\mathrm{St}(n, p)$. *Left panel: plot corresponding to Table 5.4. Right panel: plot corresponding to Table 5.5.*

We emphasize that Bryner did not use the smaller formulation on $\mathrm{St}(2p, p)$ when $p < n/2$ (see Remark 2.2 above), which makes its algorithm's complexity $\mathcal{O}(Tnp^2)$; see [13, Section 5.2]. The other algorithms considered here ([13], [30, Alg. 1], and our new SSAF algorithm) all make use of the smaller formulation on $\mathrm{St}(2p, p)$ when possible and hence they are essentially $\mathcal{O}(p^3)$, with our SSAF being comparable or even superior than [30, Alg. 1] in terms of average computation time; see Tables 5.4, 5.5, and Figure 5.1.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

As a last numerical experiment, we consider a larger value of n, namely $n = 1000$ and larger doubling values of p. The results are reported in Table 5.6, demonstrating again the competitiveness of our SSAF method in terms of both average computation time and number of iterations with respect to the existing algorithms considered here.

TABLE 5.6
Comparisons on $\mathrm{St}(1000, p)$ *for doubling values of* p *and for a prescribed* $d(X, Y) = 0.5\,\pi$. $T = 20$ *and tolerance* $\tau = 10^{-5}$. *Results are averaged over 100 experiments.*

p	Avg. comput. time (s)			Avg. no. of iterations		
	[5, Alg. 1]	[30, Alg. 1]	SSAF	[5, Alg. 1]	[30, Alg. 1]	SSAF
20	0.03897	0.00641	0.00391	4.00	3.00	5.02
40	0.09512	0.02957	0.01284	3.00	3.00	5.00
80	0.25528	0.08044	0.03969	3.00	2.00	4.00
160	0.76246	0.24119	0.13763	3.00	2.00	4.00
320	3.99810	1.07286	0.64483	3.00	2.00	4.00
640	23.36386	5.62897	2.80133	3.00	2.00	4.00

6. Conclusions and outlook. In this work we studied the shooting method, a classical numerical algorithm for solving boundary value problems, to compute the distance between two given points on the Stiefel manifold under the canonical method. We provided a shooting method for calculating geodesics on the Stiefel manifold in the sense of classical shooting methods for solving boundary value problems. The main feature of our algorithm is that we provide an approximate formula for the Fréchet derivative of the geodesic involved in our shooting method. Numerical experiments demonstrated our algorithm's performance and accuracy. We compared our algorithm to some state-of-the-art methods and showed that it is competitive with existing algorithms.

As a future outlook, an analysis of the proposed algorithm would be desirable. Moreover, we may use the knowledge gained in this work to develop a computationally cheaper algorithm. Another promising research direction is exploring the connection between shooting algorithms for computing geodesics and domain decomposition methods. Future studies will focus on these topics.

Acknowledgments. The author is grateful to Bart Vandereycken for his guidance during the author's Ph.D. thesis. Part of this work was started during the author's Ph.D. thesis at the University of Geneva, SNSF fund number 163212[4]. It was completed during the author's postdoctoral fellowship at the National Center for Theoretical Sciences in Taiwan (R.O.C.) under the NSTC grant 112-2124-M-002-009.

The author would also like to thank the anonymous referee for the many valuable comments regarding an earlier version of this paper, which considerably helped to improve the manuscript.

[4]SNSF webpage: https://data.snf.ch/grants/grant/163212

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Appendix A. List of symbols.

TABLE A.1
List of symbols.

$\mathrm{St}(n,p)$	Stiefel manifold of orthonormal n-by-p matrices
X, Y, Y_0, Y_1	Elements of $\mathrm{St}(n,p)$
I_p	The identity matrix of size p-by-p
$\mathrm{T}_X\mathrm{St}(n,p)$	Tangent space at X to the Stiefel manifold $\mathrm{St}(n,p)$
ξ^*	A tangent vector that we want to recover
Exp_X	Riemannian exponential map at X
Log_X	Riemannian logarithm map at X
$\mathcal{S}^{n-1}$	The unit sphere embedded in $\mathbb{R}^n$
$\mathrm{O}(n)$	The orthogonal group of n-by-n orthogonal matrices
$X_\perp$	An orthonormal matrix whose columns span the orthogonal complement of $\mathrm{span}(X)$
$\mathcal{S}_{\mathrm{skew}}(p)$	Space of p-by-p skew-symmetric matrices
Ω	An element of $\mathcal{S}_{\mathrm{skew}}(p)$
K	A matrix in $\mathbb{R}^{(n-p)\times p}$
P_X	The projector onto the tangent space $\mathrm{T}_X\mathrm{St}(n,p)$
$\mathcal{M}$	Generic manifold
$\mathrm{T}_x\mathcal{M}$	Tangent space at x to the manifold $\mathcal{M}$
$\langle\cdot,\cdot\rangle_x$	Inner product on the tangent space $\mathrm{T}_x\mathcal{M}$
g	Riemannian metric
$\gamma(t)$	Parametrized curve on the manifold $\mathcal{M}$
d	Riemannian distance function
$d(x,y)$	Riemannian distance between two points x and y
$\mathrm{inj}_X(\mathcal{M})$	Injectivity radius of $\mathcal{M}$ at X
$\mathrm{inj}(\mathcal{M})$	Global injectivity radius of $\mathcal{M}$
$\|\cdot\|_{\mathrm{F}}$	Frobenius norm
$\|\cdot\|_{\mathrm{c}}$	Canonical norm
$\exp_{\mathrm{m}}$	The matrix exponential
O_p	The null matrix of size p-by-p
$I_{n,p}$	The matrix $\begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}$
A	The matrix $\begin{bmatrix} \Omega & -K^\top \\ K & O_{n-p} \end{bmatrix}$
$Z_1(t)$ or $Y(t)$	A geodesic on $\mathrm{St}(n,p)$
$Z_2(t)$ or $\dot{Y}(t)$	The derivative of a geodesic
F	The nonlinear function $Z_1(1,\xi) - Y_1$
$F^{(k)}$	The nonlinear function F evaluated at iteration k
$\delta\xi^{(k)}$	The residual at iteration k in the single shooting method
$\mathrm{D}f(A)[E]$	Fréchet derivative of a matrix function f at A in the direction E

Appendix B. Approximation of the Fréchet derivative of the matrix exponential.

We recall that the Fréchet derivative of a matrix function $f\colon \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at $X \in \mathbb{C}^{n\times n}$ is the unique linear function $\mathrm{D}f(X)[\cdot]$ of the matrix $E \in \mathbb{C}^{n\times n}$ that satisfies

$$f(X+E) - f(X) - \mathrm{D}f(X)[E] = o(\|E\|).$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The mapping itself is denoted by either $\mathrm{D}f(X)[\cdot]$ or $\mathrm{D}f(X)$, while the value of the mapping for a direction E (i.e., the directional derivative) is denoted by $\mathrm{D}f(X)[E]$.

From (4.4), we have the matrix equation

$$Z_1(\xi) + Q\ \mathrm{D}\exp_{\mathrm{m}}(A(\xi))\big[\mathrm{D}A(\xi)[\delta\xi]\big] I_{n,p} - Y_1 = 0. \tag{B.1}$$

The Fréchet derivative of the matrix exponential is defined through the integral [8, (10.15)]

$$\mathrm{D}\exp_{\mathrm{m}}(A)[E] := \int_0^1 e^{A(1-s)} E\, e^{As}\, \mathrm{d}s.$$

We also have the following formula from the Taylor series of $e^{A+E} - e^A$ [8, Section 10.2]

$$\mathrm{D}\exp_{\mathrm{m}}(A)[E] = E + \frac{AE+EA}{2} + \frac{A^2E + AEA + EA^2}{3!} + \cdots.$$

As mentioned in Section 4.1, here we consider an approximation of $\mathrm{D}\exp_{\mathrm{m}}(A)[E]$ by keeping only the first two terms in the expansion, i.e.,

$$\mathrm{D}\exp_{\mathrm{m}}(A)[E] \approx E + \frac{AE+EA}{2}.$$

This truncated expansion can be used to approximate $\mathrm{D}\exp_{\mathrm{m}}(A(\xi))\big[\mathrm{D}A(\xi)[\delta\xi]\big]$ in (B.1), which yields

$$Q \cdot \big(\mathrm{D}A(\xi)[\delta\xi] + \tfrac{1}{2}(A \cdot \mathrm{D}A(\xi)[\delta\xi] + \mathrm{D}A(\xi)[\delta\xi] \cdot A)\big) \cdot I_{n,p} = Y_1 - Z_1.$$

Left-multiplying the last equation by $Q^\top$, we get

$$\big(\mathrm{D}A(\xi)[\delta\xi] + \tfrac{1}{2}(A \cdot \mathrm{D}A(\xi)[\delta\xi] + \mathrm{D}A(\xi)[\delta\xi] \cdot A)\big) \cdot I_{n,p} = Q^\top (Y_1 - Z_1)\,.$$

We emphasize that $Q^\top Z_1 = \exp_{\mathrm{m}}(A)\, I_{n,p}$ and the other term $Q^\top Y_1$ does not depend on ξ; hence, in the practical implementation of the algorithm, we compute this quantity only once. Continuing with the manipulations, we obtain

$$\begin{aligned}
\begin{bmatrix}\delta\Omega\\ \delta K\end{bmatrix} + \frac{1}{2}\left(\begin{bmatrix}\Omega & -K^\top\\ K & O\end{bmatrix}\begin{bmatrix}\delta\Omega\\ \delta K\end{bmatrix} + \begin{bmatrix}\delta\Omega & -\delta K^\top\\ \delta K & O\end{bmatrix}\begin{bmatrix}\Omega\\ K\end{bmatrix}\right) &= Q^\top(Y_1 - Z_1)\,,\\
\begin{bmatrix}\delta\Omega\\ \delta K\end{bmatrix} + \frac{1}{2}\left(\begin{bmatrix}\Omega\delta\Omega - K^\top\delta K\\ K\delta\Omega\end{bmatrix} + \begin{bmatrix}\delta\Omega\Omega - \delta K^\top K\\ \delta K\Omega\end{bmatrix}\right) &= Q^\top(Y_1 - Z_1)\,,\\
\begin{bmatrix}\delta\Omega + \frac{1}{2}\Omega\delta\Omega - \frac{1}{2}K^\top\delta K + \frac{1}{2}\delta\Omega\Omega - \frac{1}{2}\delta K^\top K\\ \delta K + \frac{1}{2}K\delta\Omega + \frac{1}{2}\delta K\Omega\end{bmatrix} &= Q^\top(Y_1 - Z_1)\,,
\end{aligned}$$

$$\begin{bmatrix}\delta\Omega + \frac{1}{2}[\Omega, \delta\Omega] - \frac{1}{2}\left(K^\top\delta K + \delta K^\top K\right)\\ \delta K + \frac{1}{2}K\delta\Omega + \frac{1}{2}\delta K\Omega\end{bmatrix} = \begin{bmatrix}Q^\top(Y_1 - Z_1)\big|_{[1:p,\,:\,]}\\ Q^\top(Y_1 - Z_1)\big|_{[p+1:n,\,:\,]}\end{bmatrix} =: \begin{bmatrix}W\\ N\end{bmatrix}. \tag{B.2}$$

From the second matrix equation, we have

$$\delta K\left(I_p + \tfrac{1}{2}\Omega\right) = N - \tfrac{1}{2}K\delta\Omega, \tag{B.3}$$

and we approximate $\left(I_p + \frac{1}{2}\Omega\right)$ with I_p, i.e.,

$$\delta K = N - \tfrac{1}{2}K\delta\Omega.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Now we insert this last equation into the first matrix equation in (B.2) to solve for $\delta\Omega$, giving

$$\begin{aligned}
\delta\Omega + \tfrac{1}{2}[\Omega, \delta\Omega] - \tfrac{1}{2}\left(K^\top(N - \tfrac{1}{2}K\delta\Omega) + (N - \tfrac{1}{2}K\delta\Omega)^\top K\right) &= W,\\
\delta\Omega + \tfrac{1}{2}\Omega\delta\Omega + \tfrac{1}{2}\delta\Omega\Omega - \tfrac{1}{2}\left(K^\top N - \tfrac{1}{2}K^\top K\delta\Omega + N^\top K - \tfrac{1}{2}\delta\Omega^\top K^\top K\right) &= W,\\
\delta\Omega + \tfrac{1}{2}\Omega\delta\Omega + \tfrac{1}{2}\delta\Omega\Omega - \tfrac{1}{2}K^\top N + \tfrac{1}{4}K^\top K\delta\Omega - \tfrac{1}{2}N^\top K + \tfrac{1}{4}\delta\Omega^\top K^\top K &= W.
\end{aligned}$$

We use the skew-symmetry $\delta\Omega^\top = -\delta\Omega$ to get rid of the transpose,

$$\delta\Omega + \tfrac{1}{2}\Omega\delta\Omega + \tfrac{1}{2}\delta\Omega\Omega + \tfrac{1}{4}K^\top K\delta\Omega - \tfrac{1}{4}\delta\Omega K^\top K = W + \tfrac{1}{2}K^\top N + \tfrac{1}{2}N^\top K,$$

and, collecting $\delta\Omega$, we obtain

$$\begin{aligned}
\delta\Omega + \tfrac{1}{2}\delta\Omega\Omega - \tfrac{1}{4}\delta\Omega K^\top K + \tfrac{1}{2}\Omega\delta\Omega + \tfrac{1}{4}K^\top K\delta\Omega &= W + \tfrac{1}{2}K^\top N + \tfrac{1}{2}N^\top K,\\
\left(I_p + \tfrac{1}{2}\Omega + \tfrac{1}{4}K^\top K\right)\delta\Omega + \delta\Omega\left(\tfrac{1}{2}\Omega - \tfrac{1}{4}K^\top K\right) &= W + \tfrac{1}{2}K^\top N + \tfrac{1}{2}N^\top K.
\end{aligned}$$

This is a Sylvester equation that can be solved with MATLAB's command `lyap` to find $\delta\Omega$. Then δK can be found using (B.3). Let the current iteration be indexed by k. Then the tangent vector is updated as

$$\xi^{(k+1)} = \xi^{(k)} + Q \cdot \begin{bmatrix} \delta\Omega^{(k)} \\ \delta K^{(k)} \end{bmatrix}.$$

REFERENCES

[1] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Riemannian geometry of Grassmann manifolds with a view on algorithmic computation*, Acta Appl. Math., 80 (2004), pp. 199–220.

[2] ———, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, 2008.

[3] P.-A. ABSIL AND J. MALICK, *Projection-like retractions on matrix manifolds*, SIAM J. Optim., 22 (2012), pp. 135–158.

[4] N. BOUMAL, *An Introduction to Optimization on Smooth Manifolds*, Cambridge University Press, Cambridge, 2023.

[5] D. BRYNER, *Endpoint geodesics on the Stiefel manifold embedded in Euclidean space*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1139–1159.

[6] H. E. ÇETINGÜL AND R. VIDAL, *Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds*, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Conference Proceedings, Los Alamitos, 2009, pp. 1896–1902.

[7] A. EDELMAN, T. A. ARIAS, AND S. T. SMITH, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[8] N. J. HIGHAM, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.

[9] C. Y. KAYA AND J. L. NOAKES, *Leapfrog for optimal control*, SIAM J. Numer. Anal., 46 (2008), pp. 2795–2817.

[10] D. G. KENDALL, D. BARDEN, T. K. CARNE, AND H. LE, *Shape and Shape Theory*, Wiley, Chichester, 1999.

[11] J. M. LEE, *Introduction to Riemannian Manifolds*, Springer, Cham, 2018.

[12] J. LI AND S. MA, *Federated learning on Riemannian manifolds*, Preprint on arXiv, 2022. https://arxiv.org/abs/2206.05668

[13] D. NGUYEN, *Closed-form geodesics and optimization for Riemannian logarithms of Stiefel and flag manifolds*, J. Optim. Theory Appl., 194 (2022), pp. 142–166.

[14] J. L. NOAKES, *A global algorithm for geodesics*, J. Austral. Math. Soc. Ser. A, 65 (1998), pp. 37–50.

[15] J. L. NOAKES AND E. ZHANG, *Finding geodesics joining given points*, Adv. Comput. Math., 48 (2022), Paper No. 50, 27 pages.

[16] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, 2nd. ed., Springer, New York, 2006.

[17] Q. RENTMEESTERS, *Algorithms for Data Fitting on Some Common Homogeneous Spaces*, Ph.D. Thesis, ICTEAM, Université Catholique de Louvain, Louvain, 2013.

[18] W. RING AND B. WIRTH, *Optimization methods on Riemannian manifolds and their application to shape space*, SIAM J. Optim., 22 (2012), pp. 596–627.

[19] A. SRIVASTAVA AND E. P. KLASSEN, *Functional and Shape Data Analysis*, Springer, New York, 2016.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[20] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer, New York, 1993.
[21] G. SUNDARAMOORTHI, A. MENNUCCI, S. SOATTO, AND A. YEZZI, *A new geometric metric in the space of curves, and applications to tracking deforming objects by prediction and filtering*, SIAM J. Imaging Sci., 4 (2011), pp. 109–145.
[22] M. SUTTI, *Riemannian Algorithms on the Stiefel and the Fixed-Rank Manifold*, Ph.D. Thesis, Faculté des Science, University of Geneva, Geneva, 2020.
[23] ———, *Shooting methods for computing geodesics on the Stiefel manifold*, Preprint on arXiv, 2023. https://arxiv.org/abs/2309.03585
[24] M. SUTTI AND B. VANDEREYCKEN, *The leapfrog algorithm as nonlinear Gauss–Seidel*, Preprint on arXiv, 2023. https://arxiv.org/abs/2010.14137
[25] P. TURAGA, A. VEERARAGHAVAN, AND R. CHELLAPPA, *Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision*, in 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Conference Proceedings, Los Alamitos, 2008, pp. 1–8.
[26] P. TURAGA, A. VEERARAGHAVAN, A. SRIVASTAVA, AND R. CHELLAPPA, *Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 2273–2286.
[27] Y.-C. WONG, *Differential geometry of Grassmann manifolds*, Proc. Natl. Acad. Sci. U.S.A., 57 (1967), pp. 589–594.
[28] M. YIN, J. GAO, AND Y. GUO, *Nonlinear low-rank representation on Stiefel manifolds*, Electron. Lett., 51 (2015), pp. 749–751.
[29] L. YOUNES, P. W. MICHOR, J. M. SHAH, AND D. B. MUMFORD, *A metric on shape space with explicit geodesics*, Atti Accad. Naz. Lincei Rend. Lincei Mat. Appl., 19 (2008), pp. 25–57.
[30] R. ZIMMERMANN, *A matrix-algebraic algorithm for the Riemannian logarithm on the Stiefel manifold under the canonical metric*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 322–342.
[31] R. ZIMMERMANN AND K. DEBRABANT, *Parametric model reduction via interpolating orthonormal bases*, in Numerical Mathematics and Advanced Applications—ENUMATH 2017, F. A. Radu, K. Kumar, I. Berre, J. M. Nordbotten, and I. S. Pop, eds., Lect. Notes Comput. Sci. Eng., 126, Springer, Cham, 2019, pp. 683–691.
[32] R. ZIMMERMANN AND K. HÜPER, *Computing the Riemannian logarithm on the Stiefel manifold: metrics, methods, and performance*, SIAM J. Matrix Anal. Appl., 43 (2022), pp. 953–980.