AN EFFICIENT ALGORITHM FOR THE RIEMANNIAN LOGARITHM ON THE STIEFEL MANIFOLD FOR A FAMILY OF RIEMANNIAN METRICS

SIMON MATAIGNE*, RALF ZIMMERMANN †, AND NINA MIOLANE‡

Abstract. Since the popularization of the Stiefel manifold for numerical applications in 1998 in a seminal paper from Edelman et al., it has been exhibited to be a key to solve many problems from optimization, statistics and machine learning. In 2021, Hüper et al. proposed a one-parameter family of Riemannian metrics on the Stiefel manifold, subsuming the well-known Euclidean and canonical metrics. Since then, several methods have been proposed to obtain a candidate for the Riemannian logarithm given any metric from the family. Most of these methods are based on the shooting method or rely on optimization approaches. For the canonical metric, Zimmermann proposed in 2017 a particularly efficient method based on a pure matrix-algebraic approach. In this paper, we derive a generalization of this algorithm that works for the one-parameter family of Riemannian metrics. The algorithm is proposed in two versions, termed backward and forward, for which we prove that it conserves the local linear convergence previously exhibited in Zimmermann's algorithm for the canonical metric.

Key words. Stiefel manifold, Riemannian logarithm, geodesic distance, Riemannian metric

MSC codes. 15B10, 15B57, 53Z50, 65B99, 15A16

1. Introduction.

The field of statistics on manifolds has experienced significant growth in recent years, driven by a multitude of applications involving manifold-valued data — such as orthogonal frames, subspaces, fixed-rank matrices, diffusion tensors or shapes — that need to undergo processes like denoising, resampling, extrapolation, compression, clustering, or classification (see, e.g., [8, 19, 20, 28]). Given a Riemannian manifold $\mathcal{M}$, both statistical and numerical applications often require the computation of distances between data points U and $\widetilde{U} \in \mathcal{M}$ [14]. Calculating this distance relies on an oracle that provides a minimal geodesic between the two points. A minimal geodesic is a curve whose length corresponds to the Riemannian distance between U and $\widetilde{U}$. Most geometries lack a known closed-form expression for the minimal geodesic between any two points. Consequently, algorithms must be developed for these geometries to yield a candidate for the minimal geodesic. The Stiefel manifold falls into this category. It has gained popularity since the publication of the seminal paper [13] by Edelman et al. This manifold finds applications in various fields, including statistics [11], optimization [13], and deep learning [15, 17]. The implementation of the Stiefel manifold is notably available in software packages such as `Geomstats` [22], `Manopt` [7], `Manopt.jl` [4].

It has been demonstrated in [36] that the Stiefel manifold, equipped with any Riemannian metric from the one-parameter family introduced in [18], features at least one minimal geodesic between any two points. However, computing *any* geodesic between two given points on the Stiefel manifold is a challenging task, known as the *geodesic endpoint problem*. Several research efforts have proposed numerical methods to address this issue [26, 29, 32, 35, 36]. Unfortunately, none of these methods guarantees

*UCLouvain, ICTEAM, Louvain-la-Neuve, Belgium. Simon Mataigne is a Research Fellow of the Fonds de la Recherche Scientifique - FNRS. simon.mataigne@uclouvain.be

†University of Southern Denmark, Department of Mathematics and Computer Science, Odense, Denmark. zimmermann@imada.sdu.dk

‡UC Santa Barbara, Electrical and Computer Engineering, Santa Barbara, CA. Nina Miolane acknowledges funding from the NSF grant 2313150. ninamiolane@ucsb.edu

the computation of a minimal geodesic, referred to as the *logarithm problem*.

The one-parameter family of metrics introduced in [18] includes the well-known Euclidean and canonical metrics [13]. In [25], it was observed that this family of metrics can be considered as metrics arising from a Cheeger deformation. Cheeger deformation metrics were first introduced in [12] and have since been used in Riemannian geometry to construct metrics with special features, e.g., non-negative curvature, or Einstein metrics. We refer to [25] for further details and additional references. Numerical algorithms for computing geodesics and Riemannian normal coordinates under this family of metrics were introduced in [36] and [26]. The case of the canonical metric allowed for special treatment, which was exploited in [36, Algorithm 4] and earlier in [35]. The associated algorithm has proven local linear convergence. In the comparative study of [36], it turned to be the most efficient and robust numerical approach. It was therefore a shortcoming that [36, Algorithm 4] was only designed to work with the canonical metric.

Original contribution. In this paper, we generalize [36, Algorithm 4] to the family of metrics introduced in [18]. The algorithm is provided in two versions, termed *backward* and *forward*. For the backward iteration, we show the local linear convergence of the algorithm and we obtain an explicit expression for the convergence rate, generalizing the one obtained in [36, Proposition 7]. However, performing backward iterations require solving a nonlinear matrix equation. To alleviate the computational costs, we introduce three types of *forward* iterations that avoid solving this nonlinear matrix equation. These forward iterations preserve the local linear convergence of the algorithm.

Reproducibility statement. All the codes written to produce results and figures are available at the address https://github.com/smataigne/StiefelLog.jl.

Organization. The paper is structured as follows. We start by recalling the necessary background on the Stiefel manifold in section 2. In section 3, we recap [36, Algorithm 4] and its convergence result. We generalize the approach to the family of metrics in section 4, where we provide a convergence analysis for the backward and forward iterations. Then, we compare the performance of the new algorithms with each other in section 5. Finally, the convergence radius and the benchmark with methods from the state of the art is carried out in section 6.

Notations. For $p > 0$, we denote the $p \times p$ identity matrix by I_p. For $n > p > 0$, we define

$$I_{n\times p} \coloneqq \begin{bmatrix} I_p \\ 0_{(n-p)\times p} \end{bmatrix} \text{ and } I_{p\times n} \coloneqq I_{n\times p}^T.$$

$\mathrm{Skew}(p)$ is the set of $p \times p$ skew-symmetric matrices ($A = -A^T$) and the orthogonal group of $p \times p$ matrices is written as

$$\mathrm{O}(p) \coloneqq \{Q \in \mathbb{R}^{p\times p} \mid Q^T Q = I_p\}.$$

The special orthogonal group $\mathrm{SO}(p)$ is the subset of matrices of $\mathrm{O}(p)$ with positive unit determinant. An orthogonal completion $U_\perp$ of $U \in \mathbb{R}^{n\times p}$ is $U_\perp \in \mathbb{R}^{n\times(n-p)}$ such that $\begin{bmatrix} U & U_\perp \end{bmatrix} \in \mathrm{O}(n)$. Throughout this paper, exp and log always denote the matrix exponential and the principal matrix logarithm. The Riemannian exponential and logarithm are written with capitals, Exp and Log. Finally, $\|\cdot\|_2$ denotes the spectral matrix norm.

2. Background on the Stiefel manifold. The main references for this section are [1, 13, 36]. Given integers $n \geq p > 0$, the Stiefel manifold of orthonormal p-frames in $\mathbb{R}^n$ is defined as

$$\mathrm{St}(n,p) \coloneqq \{U \in \mathbb{R}^{n\times p} \mid U^T U = I_p,\ n \geq p\}. \tag{2.1}$$

If $n = p$, the Stiefel manifold becomes the orthogonal group, $\mathrm{St}(n,n) = \mathrm{O}(n)$. This case allows for special treatment and is extensively studied. In particular, the minimal geodesics are known in closed form. Therefore, we restrict our considerations to $n > p$. The Stiefel manifold is a differentiable manifold of dimension $np - \frac{p(p+1)}{2}$ [1]. The tangent space of $\mathrm{St}(n,p)$ at a point U can be written as

$$T_U\mathrm{St}(n,p) = \{\Delta \in \mathbb{R}^{n\times p} \mid U^T\Delta + \Delta^T U = 0\}.$$

It follows that for all $\Delta \in T_U\mathrm{St}(n,p)$, we can write $\Delta = UA + U_\perp B$ where $A \in \mathrm{Skew}(p)$, $B \in \mathbb{R}^{(n-p)\times p}$ and $U_\perp \in \mathrm{St}(n, n-p)$ is an orthogonal complement of U. The concept of tangent space is illustrated in Fig. 1. Notice that for a given $\Delta \in T_U\mathrm{St}(n,p)$, A is uniquely defined while B depends on the chosen completion $U_\perp$. Indeed $U_\perp B = (U_\perp R)(R^T B)$ for all $R \in \mathrm{O}(n-p)$. (For the initiated reader, we mention that selecting a specific completion $U_\perp$ corresponds to lifting the tangent vector Δ to a specific horizontal space. We omit the details.)

2.1. Metrics and distances. Let $\mathcal{M}$ be a differentiable manifold. A Riemannian metric on $\mathcal{M}$ is a family $\{\langle\cdot,\cdot\rangle^x : \mathrm{T}_x\mathcal{M} \times \mathrm{T}_x\mathcal{M} \mapsto \mathbb{R}\}_{x\in\mathcal{M}}$ of symmetric positive definite bi-linear forms that depends smoothly on the location $x \in \mathcal{M}$. In practice, the input arguments of the metric encode its dependency on x, so that we can write $\langle\cdot,\cdot\rangle$ without ambiguity. Edelman et al. [13] introduced two natural metrics on the Stiefel manifold, called the *Euclidean* and the *canonical* metric, respectively. These two metrics are subsumed by the family of metrics introduced in [18]. For convenience, we use another parameterization of this family and define the β-metric[1] with $\beta > 0$ as follows. For all $\Delta, \widetilde{\Delta} \in \mathrm{T}_U\mathrm{St}(n,p)$,

$$\langle\Delta,\widetilde{\Delta}\rangle_\beta \coloneqq \mathrm{Tr}\,\Delta^T(I_n - (1-\beta)UU^T)\widetilde{\Delta} = \beta\,\mathrm{Tr}\,A^T\widetilde{A} + \mathrm{Tr}\,B^T\widetilde{B}.$$

The Euclidean and canonical metrics correspond to $\beta = 1$ and $\beta = \frac{1}{2}$ respectively. Therefore, we are particularly interested in $\beta \in \left[\frac{1}{2}, 1\right]$ and our experiments will focus on this interval. The Euclidean metric is inherited from the ambient Euclidean space $\mathbb{R}^{n\times p}$ while the canonical metric is inherited from the quotient structure $\mathrm{St}(n,p) = \mathrm{SO}(n)/\mathrm{SO}(n-p)$ [13]. The norm induced by any β-metric is $\|\Delta\|_\beta = \sqrt{\langle\Delta,\Delta\rangle_\beta}$, and the length of a continuously differentiable curve $\gamma : [0,1] \mapsto \mathrm{St}(n,p)$ is given by

$$l_\beta(\gamma) = \int_0^1 \|\dot\gamma(t)\|_\beta\ \mathrm{d}t, \tag{2.2}$$

where $\dot\gamma$ denotes the time derivative of γ. For all $U, \widetilde{U} \in \mathrm{St}(n,p)$, we obtain an induced distance function [30, Proposition 1.1] (also termed *Riemannian distance*) defined as

$$d_\beta(U,\widetilde{U}) = \inf\{l_\beta(\gamma) \mid \gamma(0) = U,\ \gamma(1) = \widetilde{U}\}. \tag{2.3}$$

Since β-geodesics exist for all times [18], the Hopf-Rinow theorem ensures that a curve achieves the minimal distance: the "inf" in (2.3) is a "min" when $\mathcal{M} = \mathrm{St}(n,p)$.

[1] [2] and [26] also used this more convenient parameterization.

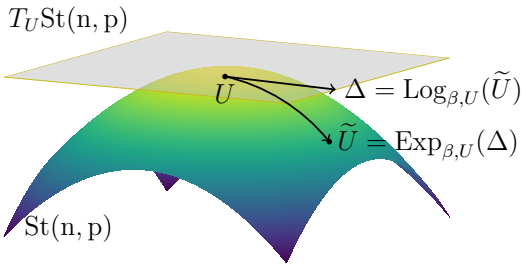

FIG. 1. *Conceptual illustration of the Stiefel manifold* $\mathrm{St}(n,p)$*, the tangent space* $T_U\mathrm{St}(n,p)$*, the exponential map* $\mathrm{Exp}_{\beta,U}(\Delta)$ *and the logarithmic map* $\mathrm{Log}_{\beta,U}(\widetilde{U})$ *(see Section subsection 2.2).*

Riemannian metrics act on tangent vectors, while the Riemannian distance is for points on the manifold. Since $\mathrm{St}(n,p)$ is embedded in $\mathbb{R}^{n\times p}$, the distance between $U, \tilde{U} \in \mathrm{St}(n,p)$ can also be measured in the ambient space. An upper bound for this distance in the Frobenius norm is

$$\|U - \tilde{U}\|_F \leq \|U\|_F + \|\tilde{U}\|_F = 2\sqrt{\mathrm{Tr}(I_p)} = 2\sqrt{p},$$

and is referred to as the *Frobenius diameter* of the Stiefel manifold.

2.2. The Riemannian exponential and logarithm. Consider $\mathrm{St}(n,p)$ endowed with $\langle\cdot,\cdot\rangle_\beta$, written $(\mathrm{St}(n,p), \langle\cdot,\cdot\rangle_\beta)$. Zimmermann and Hüper [36] showed that the *Riemannian exponential* $\mathrm{Exp}_{\beta,U} : T_U\mathrm{St}(n,p) \mapsto \mathrm{St}(n,p)$, i.e., the function that maps $\Delta \in T_U\mathrm{St}(n,p)$ to the point reached at unit time by the geodesic with starting point U and initial velocity Δ (see, e.g., [30, Chap. II, Sec. 2]), can be expressed according to Theorem 2.1. The mode of operation of the Riemannian exponential is illustrated in Fig. 1.

Theorem 2.1. THE RIEMANNIAN EXPONENTIAL [36, Equation 10]. *For all* $U \in \mathrm{St}(n,p)$ *and* $\Delta \in T_U\mathrm{St}(n,p)$*, we have*

$$\mathrm{Exp}_{\beta,U}(\Delta) = \begin{bmatrix} U & Q \end{bmatrix} \exp\left(\begin{bmatrix} 2\beta A & -B^T \\ B & 0 \end{bmatrix}\right) I_{n\times p} \exp\left((1-2\beta)A\right), \tag{2.4}$$

where $A = U^T\Delta \in \mathrm{Skew}(p)$ *and* $QB = (I - UU^T)\Delta \in \mathbb{R}^{n\times p}$ *is any matrix decomposition where* $Q \in \mathrm{St}(n, n-p)$ *with* $Q^TU = 0$ *and* $B \in \mathbb{R}^{(n-p)\times p}$.

If $n > 2p$, we can always reduce the decomposition of Δ such that $Q \in \mathrm{St}(n,p)$ and $B \in \mathbb{R}^{p\times p}$ [29, 36]. The first matrix exponential of (2.4) belongs then to $\mathbb{R}^{2p\times 2p}$ instead of $\mathbb{R}^{n\times n}$. This property yields significant computational savings if $n \gg p$. Given $U, \widetilde{U} \in \mathrm{St}(n,p)$, the Riemannian logarithm $\mathrm{Log}_{\beta,U}(\widetilde{U})$ is the function returning the set of all minimal-norm tangent vectors $\Delta \in T_U\mathrm{St}(n,p)$ such that $\mathrm{Exp}_{\beta,U}(\Delta) = \widetilde{U}$. Locally, the Riemannian logarithm is the inverse of the exponential.

Problem 2.2. THE LOGARITHM PROBLEM. *Let* $U, \widetilde{U} \in \mathrm{St}(n,p)$ *and* $\beta > 0$*. The Riemannian logarithm* $\mathrm{Log}_{\beta,U}(\widetilde{U})$ *returns all* $\Delta \in T_U\mathrm{St}(n,p)$ *such that*

$$\mathrm{Exp}_{\beta,U}(\Delta) = \widetilde{U} \text{ and } \|\Delta\|_\beta = d_\beta(U, \widetilde{U}).$$

The curves $[0,1] \ni t \mapsto \mathrm{Exp}_{\beta,U}(t\Delta)$ *are then called* minimal geodesics.

Because the Stiefel manifold is complete, the Hopf-Rinow theorem [10, Chap. 7, Thm. 2.8] ensures the existence of at least one minimal geodesic between any two

points on the Stiefel manifold. However, on compact Riemannian manifolds, every geodesic has a designated *cut time* t^* [30]: the geodesic stays minimal as long as $t \leq t^*$, not beyond. The point $\mathrm{Exp}_{\beta,U}(t^*\Delta)$ is termed the *cut point*. Consequently, when a geodesic between two points is established, the shooting direction is a Riemannian logarithm if and only if the destination is reached before the cut point. In [36, Thm. 3], it is emphasized that the geodesic endpoint problem on the Stiefel manifold boils down to solving a nonlinear matrix problem. Very recent works [2, 31] proposed a thin interval for the injectivity radius $\mathrm{inj}_{\mathrm{St}(n,p)}$, i.e., the distance to the nearest cut point. In particular for $\beta = \frac{1}{2}$, they showed that $0.894\pi < \mathrm{inj}_{\mathrm{St}(n,p)} < 0.914\pi$. Within the injectivity radius, any solution to the geodesic endpoint problem gives the unique solution to Problem 2.2, namely, *the* Riemannian logarithm.

3. State of the art. [29, 35] introduced an efficient method to compute a geodesic between any two given points $U, \widetilde{U} \in \mathrm{St}(n,p)$, but only in the case of the canonical metric ($\beta = \frac{1}{2}$) and $n \geq 2p$. The algorithm was enhanced in [36] to obtain a better convergence rate. Its pseudo-code is given by Algorithm 3.1. Previous experiments [36, Table 1] highlighted the better performance of Algorithm 3.1 compared to the shooting method, introduced first in [9]. For the β-metric, [26] also proposed a L-BFGS method inspired from the shooting method. The L-BFGS method was however shown to converge slower than the shooting method [26, Table 2]. This observation provides a strong incentive to generalize Algorithm 3.1 to the parameterized family of β-metrics.

Algorithm 3.1 finds a vector Δ satisfying $\widetilde{U} = \mathrm{Exp}_{\beta,U}(\Delta)$ when $\beta = \frac{1}{2}$. In this case, Problem 2.2 simplifies and consists of finding matrices $A \in \mathrm{Skew}(p)$ and $B \in \mathbb{R}^{p\times p}$ such that

$$\begin{bmatrix} M \\ N \end{bmatrix} = \exp \begin{bmatrix} A & -B^T \\ B & 0 \end{bmatrix} I_{2p\times p}, \tag{3.1}$$

where $M := U^T\widetilde{U}$, $N := Q^T\widetilde{U}$ with $Q \in \mathrm{St}(n,p)$ such that $\mathrm{col}([U\ \widetilde{U}]) \subseteq \mathrm{col}([U\ Q])$ and $Q^TU = 0$. Equation (3.1) is solved iteratively by computing a sequence of matrices $\{A_k, B_k, C_k\}_{k\in\mathbb{N}}$ by

$$\begin{bmatrix} A_k & -B_k^T \\ B_k & C_k \end{bmatrix} := \log V_k \in \mathrm{Skew}(2p) \text{ with } V_k := \begin{bmatrix} M & O_k \\ N & P_k \end{bmatrix} \in \mathrm{SO}(2p). \tag{3.2}$$

O_0, P_0 are chosen such that $V_0 \in \mathrm{SO}(2p)$ and V_k is updated to have $\lim_{k\to\infty} \|C_k\|_2 = 0$. Theorem 3.2 below shows that taking $V_{k+1} = V_k \begin{bmatrix} I_p & 0 \\ 0 & \exp(\Gamma_k) \end{bmatrix}$, where Γ_k solves the *Sylvester equation*

$$S\Gamma_k + \Gamma_k S = C_k \text{ with } S := \frac{1}{12} B_k B_k^T - \frac{I_p}{2}, \tag{3.3}$$

yields local linear convergence, i.e., $\|C_{k+1}\|_2 \leq a\|C_k\|_2 + \mathcal{O}(\|C_k\|_2^2)$ with $a < 1$. Ultimately, Algorithm 3.1 outputs a geodesic $\gamma : [0,1] \mapsto \mathrm{St}(n,p)$ between U and $\widetilde{U}$ defined by

$$\gamma(t) := \mathrm{Exp}_{\frac{1}{2},U}(t\Delta) = \begin{bmatrix} U & Q \end{bmatrix} \exp\left(t \begin{bmatrix} A_\infty & -B_\infty^T \\ B_\infty & 0 \end{bmatrix} \right) I_{2p\times p}. \tag{3.4}$$

Algorithm 3.1 possesses two strengths. First, $\{A_k, B_k, C_k\}$ provides a curve between U and $\widetilde{U}$ at any iteration k, which becomes a (numerical) geodesic at convergence

when $\|C_k\|_2 < \varepsilon$ for some tolerance $\varepsilon > 0$. Second, unlike the shooting method, no time-discretization of the geodesic is needed. This is a strong advantage since i) the discretization is computationally expensive: a Riemannian exponential and an approximate parallel transport have to be computed at every discretized point along the geodesic and ii) the level of discretization is a user-parameter that has to be tuned, or guessed. On the other hand, the shooting method involves computing the matrix exponential of skew-symmetric matrices. At this day, it can be done more efficiently — such as in `SkewLinearAlgebra.jl` — than computing the matrix logarithm in (3.2). Nonetheless [36, Table 1] observed that Algorithm 3.1 runs faster due to its better convergence rate. It was thus a deficiency that the most efficient method among the considered ones could not be generalized to the β-metrics. We close this gap in the next sections.

Algorithm 3.1 [36, Algorithm 4] Improved algebraic Stiefel logarithm for the canonical metric ($\beta = \frac{1}{2}$).

1: **INPUT:** Given $U, \widetilde{U} \in \mathrm{St}(n,p)$, $n \geq 2p$ and $\varepsilon > 0$, compute:
2: Set $M = U^T\widetilde{U}$.
3: Compute $\widehat{Q}\widehat{N} = (I - UU^T)\widetilde{U}$. #where $\widehat{Q} \in \mathrm{St}(n,p)$ and $\widehat{Q}^T U = 0$.
4: Build $V_0 = \begin{bmatrix} M & O_0 \\ N & P_0 \end{bmatrix} \in \mathrm{SO}(2p)$, $Q \in \mathrm{St}(n,p)$. #see Appendix B for N, O_0, P_0, Q.
5: **for** $k = 0, 1, \ldots$ **do**
6: Compute $\begin{bmatrix} A_k & -B_k^T \\ B_k & C_k \end{bmatrix} = \log V_k$.
7: **if** $\|C_k\| \leq \varepsilon$ **then**
8: Break.
9: **end if**
10: Compute Γ_k solving $\Gamma_k S + S\Gamma_k = C_k$ where $S \coloneqq \frac{1}{12} B_k B_k^T - \frac{I_p}{2}$.
11: Update $V_{k+1} = V_k \begin{bmatrix} I_p & 0 \\ 0 & \exp(\Gamma_k) \end{bmatrix}$.
12: **end for**
13: **return** $\Delta = UA_k + QB_k \in T_U\mathrm{St}(n,p)$.

Remark 3.1. In line 3 of Algorithm 3.1, [35] proposed to compute a thin QR decomposition $\widehat{Q}\widehat{N} = (I - UU^T)\widetilde{U}$. It is without ambiguity if $(I - UU^T)\widetilde{U}$ has full column rank. However, if $(I - UU^T)\widetilde{U}$ is not full column rank, it is important to ensure $\widehat{Q} \in \mathrm{St}(n,p)$ with $\widehat{Q}^T U = 0$. Otherwise, Algorithm 3.1 may fail to find any $\Delta \in \mathrm{Log}_{\beta,U}(\widetilde{U})$. Indeed, there are provable cases where

$$\mathrm{rank}((I - UU^T)\Delta) > \mathrm{rank}((I - UU^T)\widetilde{U}). \tag{3.5}$$

We show in Appendix A that (3.5) can only happen if $\widetilde{U}$ belongs to the cut locus of U.

The linear convergence result for Algorithm 3.1 is given by Theorem 3.2. In subsection 4.3, we first generalize the algorithm to work with the parametric family of metrics in form of Algorithm 4.1. The associated generalization of the convergence statement Theorem 3.2 is then Theorem 4.3, which reduces to the original result for $\beta = \frac{1}{2}$.

Theorem 3.2. [36, Prop. 7] Given $U, \widetilde{U} \in \mathrm{St}(n,p)$ $(U \neq \widetilde{U})$ such that Algorithm 3.1 converges. Assume further that there is $0 < \delta < 1$ such that for $\log(V_k) = \begin{bmatrix} A_k & -B_k^T \\ B_k & C_k \end{bmatrix}$, it holds $\|\log(V_k)\|_2 < \delta$ throughout the algorithm's iteration loop. Then, for k large enough, it holds

$$\|C_{k+1}\|_2 \leq \frac{6}{6-\delta^2}\frac{\delta^4}{1-\delta}\|C_k\|_2 + \mathcal{O}(\|C_k\|_2^2). \tag{3.6}$$

Theorem 3.2 highlights that the closer U and $\widetilde{U}$, the faster Algorithm 3.1 should converge. This property was numerically confirmed in [35, Sec. 5.3].

4. The generalization of Algorithm 3.1. We propose a new approach to generalize Algorithm 3.1 to all the β-metrics. Still, we assume $n \geq 2p$ which matches the '$n \gg p$' setting of practical big-data applications.

When $\beta \neq \frac{1}{2}$, a generalization of Algorithm 3.1 consists of finding a sequence of matrices $\{A_k, B_k, C_k\}_{k\in\mathbb{N}}$ with $A_k, C_k \in \mathrm{Skew}(p)$, $B_k \in \mathbb{R}^{p\times p}$ such that

$$\begin{bmatrix} M \\ N \end{bmatrix} = \exp\begin{bmatrix} 2\beta A_k & -B_k^T \\ B_k & C_k \end{bmatrix} I_{2p\times p} \exp((1-2\beta)A_k), \tag{4.1}$$

where $M \coloneqq U^T\widetilde{U}$, $N \coloneqq Q^T\widetilde{U}$ $(Q \in \mathrm{St}(n,p),\ Q^TU = 0)$ and $\lim_{k\to\infty}\|C_k\|_2 = 0$. Since A_k appears twice in (4.1), it is a challenging non-linear equation to solve. Moreover, straightforward usage of Newton's method is very expensive [36]. However, the equation can be tackled by constructing a sequence

$$\begin{bmatrix} 2\beta A_k & -B_k^T \\ B_k & C_k \end{bmatrix} \coloneqq \log\left(V_k \begin{bmatrix} \exp(-(1-2\beta)\widehat{A}_k) & 0 \\ 0 & I_p \end{bmatrix}\right), \tag{4.2}$$

where $\widehat{A}_k$ is a consistent approximation, i.e., $\lim_{k\to\infty}\|\widehat{A}_k - A_k\|_2 = 0$. V_k follows its definition from (3.2). The matrix sequence $\{V_k\}_{k\in\mathbb{N}}$ is chosen such that $\lim_{k\to\infty}\|C_k\|_2 = 0$. If one knows $\widehat{A}_k = A_k$, we term it a *backward iteration*. This is too expensive in practice and we rather perform a *forward iteration* by selecting $\widehat{A}_k$ as an approximation of A_k. Many different ideas can be exploited, we propose three of them. The simplest one is $\widehat{A}_k \coloneqq A_{k-1}$. We term it a *fixed forward iteration*. To improve this approximation, we can solve (4.1) for A_k approximately and at low computational cost, based on our knowledge of A_{k-1}. We term this a *pseudo-backward iteration*. Finally, we can improve the fixed forward approximation by taking $\widehat{A}_k \coloneqq A_{k-1} + hQ_{k-1}(A_{k-1} - \widehat{A}_{k-1})Q_{k-1}^T$ for some $h \in \mathbb{R}$ and $Q_{k-1} \in \mathrm{O}(p)$. We term this an *accelerated forward iteration*. All three possibilities are investigated in subsection 4.4. Algorithm 4.1 is a pseudo-code of the generalization of Algorithm 3.1 to the β-metric.

Remark 4.1. At line 3 of Algorithm 4.1, Remark 3.1 still holds.

4.1. The fundamental equation of Algorithm 4.1. We write explicitly how an iteration of the algorithm relates to the previous one. This will facilitate our analysis of the algorithm. By line 7 of Algorithm 4.1, it holds that

$$V_k = \exp\left(\begin{bmatrix} 2\beta A_k & -B_k^T \\ B_k & C_k \end{bmatrix}\right)\exp\left(\begin{bmatrix} (1-2\beta)\widehat{A}_k & 0 \\ 0 & 0 \end{bmatrix}\right).$$

Algorithm 4.1 Improved algebraic Stiefel logarithm for the β-metric family.

1: **INPUT:** Given $U, \widetilde{U} \in \mathrm{St}(n,p)$, $n \geq 2p$ and $\varepsilon > 0$, compute:
2: Set $M = U^T \widetilde{U}$.
3: Compute $\widehat{Q}\widehat{N} = (I - UU^T)\widetilde{U}$. #where $\widehat{Q} \in \mathrm{St}(n,p)$ and $\widehat{Q}^T U = 0$.
4: Build $V_0 = \begin{bmatrix} M & O_0 \\ N & P_0 \end{bmatrix} \in \mathrm{SO}(2p)$, $Q \in \mathrm{St}(n,p)$. #see Appendix B for N, O_0, P_0, Q.
5: **for** $k = 0, 1, \ldots$ **do**
6: Take an approximation $\widehat{A}_k \approx A_k$. #see subsection 4.4.
7: Compute $\begin{bmatrix} 2\beta A_k & -B_k^T \\ B_k & C_k \end{bmatrix} = \log\left(V_k \begin{bmatrix} \exp(-(1-2\beta)\widehat{A}_k) & 0 \\ 0 & I_p \end{bmatrix} \right)$.
8: **if** $\|C_k\| + \|\widehat{A}_k - A_k\| \leq \varepsilon$ **then**
9: Break.
10: **end if**
11: Compute Γ_k solving $\Gamma_k S + S\Gamma_k = C_k$ where $S := \frac{1}{12} B_k B_k^T - \frac{I_p}{2}$.
12: Update $V_{k+1} = V_k \begin{bmatrix} I_p & 0 \\ 0 & \exp(\Gamma_k) \end{bmatrix}$.
13: **end for**
14: **return** $\Delta = UA_k + QB_k \in T_U \mathrm{St}(n,p)$.

Introducing line 12 of Algorithm 4.1, i.e., $V_{k+1} = V_k \left[\begin{smallmatrix} I_p & 0 \\ 0 & \exp(\Gamma_k) \end{smallmatrix} \right]$, we obtain

$$\begin{bmatrix} 2\beta A_{k+1} & -B_{k+1}^T \\ B_{k+1} & C_{k+1} \end{bmatrix} = \log\left(\exp\left(\begin{bmatrix} 2\beta A_k & -B_k^T \\ B_k & C_k \end{bmatrix} \right) \exp\left(\begin{bmatrix} \Theta_k & 0 \\ 0 & \Gamma_k \end{bmatrix} \right) \right), \tag{4.3}$$

where Θ_k is defined by

$$\Theta_k := \log\left(\exp((1-2\beta)\widehat{A}_k) \exp(-(1-2\beta)\widehat{A}_{k+1}) \right). \tag{4.4}$$

The loop generated by (4.3) is driven by two parameters: Γ_k and $\widehat{A}_k$. The next sections are dedicated to the study of the possible choices for Γ_k and $\widehat{A}_k$.

4.2. BCH formulas for Γ_k and $\widehat{A}_k$. Algorithm 3.1 chose Γ_k using the Baker-Campbell-Hausdorff (BCH) formula. For X, Y in the Lie algebra (e.g., $\mathrm{Skew}(n)$) of a Lie group (e.g., $\mathrm{SO}(n)$), the BCH formula gives

$$\log(\exp(X)\exp(Y)) = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}[X-Y,[X,Y]] + \text{H.O.T}(4), \tag{4.5}$$

where $[X,Y] := XY - YX$ is called the *Lie bracket* or *commutator*. H.O.T(4) stands for "Higher Order Terms of 4th order". The term 'order' has to be read with care. It refers to terms in X, Y of combined order 4 or higher. For example, X^3Y and $XYXY$ are such 4th-order terms.

Letting $Z := \log(\exp(X)\exp(Y))$, [34, Thm. 1] showed that (4.5) was convergent for $\|X\|_2, \|Y\|_2 \leq \mu < 1$. In the upcoming Condition 4.4, we state locality assumptions that are always sufficient for the convergence of (4.5), namely conditions that yield $\|X\|_2, \|Y\|_2 \leq \mu < 1$. In view of the fundamental equation (4.3), we define

$$X_k = \begin{bmatrix} 2\beta A_k & -B_k^T \\ B_k & C_k \end{bmatrix} \text{ and } Y_k = \begin{bmatrix} \Theta_k & 0 \\ 0 & \Gamma_k \end{bmatrix}. \tag{4.6}$$

The BCH series expansion of the $p \times p$ bottom-right block up to fifth order of (4.5) yields

$$C_{k+1} = C_k + \Gamma_k - \frac{1}{12}\left(\Gamma_k B_k B_k^T + B_k B_k^T \Gamma_k\right) \tag{4.7}$$

$$+ \mathcal{O}(\|\Gamma_k\|_2^i \|C_k\|_2^j) \tag{4.8}$$

$$+ \frac{1}{6} B_k \Theta_k B_k^T \tag{4.9}$$

$$- \frac{1}{12}\left(B_k \Theta_k B_k^T \Gamma_k - \Gamma_k B_k \Theta_k B_k^T\right) \tag{4.10}$$

$$+ \text{H.O.T}_C(5), \tag{4.11}$$

with $i, j \geq 1$ in (4.8). The objective pursued in [36] to prove the linear convergence of Algorithm 3.1 was to show that $\|C_{k+1}\|_2 \leq a\|C_k\|_2 + \mathcal{O}(\|C_k\|_2^2)$ with $0 < a < 1$. The terms (4.7) and (4.8) are the same as in [36]. The terms (4.7) can thus be cancelled similarly by taking Γ_k as the solution to the *Sylvester equation* (3.3). Assuming $\|B_k\|_2, \|C_k\|_2 < \delta$ with $\delta < 1$, Γ_k satisfies

$$\|\Gamma_k\|_2 \leq \frac{6}{6 - \delta^2}\|C_k\|_2 \text{ (see [36, p.967])}. \tag{4.12}$$

Hence, (4.8) turns out to be $\mathcal{O}(\|C_k\|_2^2)$, and is thus neglected. The commonalities with [36] end here since the other terms are new or different. We need a bound on (4.9), (4.10) and (4.11) in terms of $\|C_k\|_2$. From our experience, it would be a mistake to cancel (4.9) and/or (4.10) with Γ_k because of the negative influence it would have on (4.11). Hence, the challenge is to choose $\widehat{A}_k$ where $\|\Theta_k\|_2$ converges linearly to 0 when $\|C_k\|_2$ does. Theorem 4.3 shows that the *backward iteration*, i.e., taking $\widehat{A}_k = A_k$, verifies this property. We also propose convergent *forward iterations* in subsection 4.4 based on Theorem 4.3.

4.3. Linear convergence of the backward iteration. In this section, we generalize the analysis that was done for Algorithm 3.1 in [36], but for Algorithm 4.1. That is, assuming convergence, we prove the local linear convergence of Algorithm 4.1 implemented with backward iterations. In subsection 4.4, we generalize the convergence result to forward iterations. Theorem 4.3 asks for a locality assumption as it was done in Theorem 3.2 with $\left\| \begin{bmatrix} A_k & -B_k^T \\ B_k & C_k \end{bmatrix} \right\|_2 < \delta$. Given the *fundamental equation* (4.3), it would be natural for us to consider $\left\| \begin{bmatrix} 2\beta A_k & -B_k^T \\ B_k & C_k \end{bmatrix} \right\|_2 < \delta$. It turns out that we need to bound both aforementioned matrices. The first option yields an easier expression for Theorem 4.3 since it allows to write $\|A_k\|_2 < \delta$ instead of $\|A_k\|_2 < \frac{\delta}{2\beta}$. Lemma C.1 shows that these two choices are equivalent up to a multiplicative factor. The initial choice is thus arbitrary in view of the asymptotic behavior.

Remark 4.2. Important efforts have been engaged in [35] to obtain a sufficient condition on $\|\widetilde{U} - U\|_{\text{F}}$ for the convergence of Algorithm 3.1. This led to $\|\widetilde{U} - U\|_{\text{F}} < 0.091$ [35, Thm. 4.1]. This value is not satisfying because it is not representative of the condition observed in practice. We want a notion of radius of convergence that is probabilistic and scales with the size of $\text{St}(n, p)$, e.g., $2\sqrt{p}$, the Frobenius diameter. We propose the following approach. We obtain a theoretical linear rate of convergence for Algorithm 4.1 in Theorem 4.3 at the cost of feasibility assumptions (Condition 4.4) and then we investigate the convergence radius numerically in subsection 6.1.

Theorem 4.3 is a generalization of Theorem 3.2, borrowing voluntarily its format, and reducing to it when $\beta = \frac{1}{2}$ (canonical metric).

Theorem 4.3. Given $\beta > \frac{1}{4}$, $U, \widetilde{U} \in \mathrm{St}(n,p)$ $(U \neq \widetilde{U})$ and Algorithm 4.1 with $\widehat{A}_k = A_k$ in step 6. Assume there is k such that X_k, Y_k from (4.6) satisfy $\|Y_k\|_2 \leq \|X_k\|_2$. Assume further that there is $\delta > 0$ satisfying Condition 4.4 and such that for $L_k := \begin{bmatrix} A_k & -B_k^T \\ B_k & C_k \end{bmatrix}$ it holds $\|L_k\|_2 < \delta$ throughout the algorithm's iteration loop. Then, it holds that

$$\|C_{k+1}\|_2 \leq \underbrace{\left(\frac{\eta\delta^4}{6\xi(6-\delta^2)} + \left(1+\frac{\eta}{\xi}\right) \frac{\kappa\alpha}{1-\frac{\eta\alpha}{\xi}} \right)}_{\mathcal{O}(\delta^4)} \|C_k\|_2 + \mathcal{O}(\|C_k\|_2^2), \tag{4.13}$$

where the constant factors are defined by

$$\begin{aligned} \tau &:= 1-2\beta, & \eta &:= |\tau| \left(1 + \delta|\tau| + \frac{2}{3}\delta^2|\tau|^2 - \delta^2|\tau|^2 \log\left(1-2\delta|\tau|\right)\right), \\ \alpha &:= \frac{\delta^4(1+|\tau|)^4}{1-\delta(1+|\tau|)}, & \kappa &:= \max\left(\frac{6}{6-\delta^2}, \frac{\eta\delta^2}{\xi(6-\delta^2)}\right), \\ \xi &:= 2\beta - \eta\left(1 + 2\beta\delta + \frac{4\beta^2}{3}\delta^2 + \frac{\delta^2}{6} + \frac{\delta^3}{6-\delta^2}\right). \end{aligned}$$

Proof. The proof consists of bounding (4.9), (4.10) and (4.11) in terms of $\|C_k\|_2$. Although it is quite tedious, we demonstrate that it can be done. Recall that $\|L_k\|_2 < \delta$ implies that $\|A_k\|_2, \|B_k\|_2, \|C_k\|_2 < \delta$. This proof involves the series expansions of several matrices with respective higher order terms. To avoid ambiguity, we append subscripts to "H.O.T" to distinguish them. There will be various conditions for the well-definedness of the parameters involved. These are only listed collectively in Condition 4.4 after the proof, as they would seem unmotivated at this point. A first milestone is to bound $\|\Theta_k\|_2$ in terms of $\|C_k\|_2$. The BCH series expansion of Θ_k, based on (4.4), leads to

$$\Theta_k = \tau(A_k - A_{k+1}) - \frac{\tau^2}{2}[A_k, A_{k+1}] - \frac{\tau^3}{12}[A_k + A_{k+1}, [A_k, A_{k+1}]] + \mathrm{H.O.T}_\Theta(4). \tag{4.14}$$

By Lemma C.2, we have $\|\mathrm{H.O.T}_\Theta(4)\|_2 \leq |\tau|(|\tau|\delta)^2 \log\left(\frac{1}{1-2\delta|\tau|}\right) \|A_k - A_{k+1}\|_2 =: |\tau|\zeta\|A_k - A_{k+1}\|_2$. If we take the norm on both sides and mind that $[A_k, A_{k+1}] = [A_k, A_{k+1} - A_k]$, we get a bound on $\|\Theta_k\|_2$ in terms of $\|A_k - A_{k+1}\|_2$:

$$\begin{aligned} \|\Theta_k\|_2 &\leq |\tau|(1+\zeta)\|A_k - A_{k+1}\|_2 + \left(\frac{|\tau|^2}{2} + \frac{|\tau|^3}{6}\|A_k + A_{k+1}\|_2\right) \|[A_k, A_{k+1}]\|_2 \\ &\leq |\tau|(1+\zeta)\|A_k - A_{k+1}\|_2 + \left(\frac{|\tau|^2}{2} + \frac{|\tau|^3}{6} 2\delta\right) 2\|A_k\|_2 \|A_k - A_{k+1}\|_2 \\ &\leq \eta\|A_k - A_{k+1}\|_2, \end{aligned} \tag{4.15}$$

where $\eta := |\tau|(1 + \delta|\tau| + \frac{2}{3}\delta^2|\tau|^2 + \zeta)$. Now, we leverage the top-left $p \times p$ block of (4.5) to bound $\|A_k - A_{k+1}\|_2$ in terms of $\|C_k\|_2$. This is done in Lemma C.3 and

yields

$$2\beta\|A_k - A_{k+1}\|_2 \leq \eta\left(1 + 2\beta\delta + \frac{4\beta^2}{3}\delta^2 + \frac{\delta^2}{6} + \frac{\delta^3}{6-\delta^2}\right)\|A_k - A_{k+1}\|_2 + \frac{\delta^2}{6-\delta^2}\|C_k\|_2 + \|\text{H.O.T}_A(5)\|_2 + \mathcal{O}(\|A_k - A_{k+1}\|_2^2). \tag{4.16}$$

By Condition 4.4, we have $\xi := 2\beta - \eta\left(1 + 2\beta\delta + \frac{4\beta^2}{3}\delta^2 + \frac{\delta^2}{6} + \frac{\delta^3}{6-\delta^2}\right) > 0$, which leads to

$$\|A_k - A_{k+1}\|_2 \leq \frac{\delta^2}{\xi(6-\delta^2)}\|C_k\|_2 + \frac{1}{\xi}\|\text{H.O.T}_A(5)\|_2 + \mathcal{O}(\|A_k - A_{k+1}\|_2^2). \tag{4.17}$$

By inserting (4.17) into $\mathcal{O}(\|A_k - A_{k+1}\|_2^2)$, it follows that

$$\mathcal{O}(\|A_k - A_{k+1}\|_2^2) \in \mathcal{O}(\|C_k\|_2^2, \|\text{H.O.T}_A(5)\|_2^2, \|C_k\|_2\|\text{H.O.T}_A(5)\|_2).$$

We introduce the short-hand notation Ψ to represent all $\mathcal{O}(\|A_k - A_{k+1}\|_2^2)$ terms. In particular, we consider $\alpha\Psi = \Psi$ for all $\alpha > 0$. We will obtain further that $\Psi \in \mathcal{O}(\|C_k\|_2^2)$. Inserting (4.17) into (4.15) finally yields

$$\|\Theta_k\|_2 \leq \frac{\eta\delta^2}{\xi(6-\delta^2)}\|C_k\|_2 + \frac{\eta}{\xi}\|\text{H.O.T}_A(5)\|_2 + \Psi. \tag{4.18}$$

Only the higher order terms are still to be eliminated. Notice that by the properties of $\|\cdot\|_2$, the block-expansions $\|\text{H.O.T}_A(5)\|_2$ from (4.18) and $\|\text{H.O.T}_C(5)\|_2$ from (4.11) are both bounded from above by $\|\text{H.O.T}(5)\|_2$, the higher order terms of the complete series expansion of $\log(\exp(X_k)\exp(Y_k))$ from (4.6). By definition of the BCH series expansion of X_k and Y_k, we have

$$\|\text{H.O.T}(5)\|_2 \leq \sum_{l=5}^{\infty}\|z_l(X_k, Y_k)\|_2, \tag{4.19}$$

where $z_l(X_k, Y_k)$ is the sum over all words of length l in the alphabet $\{X_k, Y_k\}$ multiplied by their respective Goldberg coefficient of order l [16]. The goal is to obtain a relation between $\|X_k\|_2, \|Y_k\|_2$ and $\|C_k\|_2$. By (4.12) and (4.18), it holds

$$\begin{aligned}\|Y_k\|_2 &= \max(\|\Gamma_k\|_2, \|\Theta_k\|_2)\\ &\leq \max\left(\frac{6}{6-\delta^2}, \frac{\eta\delta^2}{\xi(6-\delta^2)}\right)\|C_k\|_2 + \frac{\eta}{\xi}\|\text{H.O.T}_A(5)\|_2 + \Psi.\end{aligned}$$

It is now convenient to leverage $\|Y_k\|_2 \leq \|X_k\|_2$ for k large enough because it allows to simplify (4.19) greatly. It was already used in [36] for the proof of Theorem 3.2. It is a valid hypothesis since $U \neq \widetilde{U}$ must lead to $0 = \|Y_\infty\|_2 < \|X_\infty\|_2$ upon convergence. By Lemma C.1, we have $\|X_k\|_2 \leq (1+|\tau|)\|L_k\|_2$, where we recall that $L_k = \begin{bmatrix} A_k & -B_k^T \\ B_k & C_k \end{bmatrix}$. Defining $\kappa := \max\left(\frac{6}{6-\delta^2}, \frac{\eta\delta^2}{\xi(6-\delta^2)}\right)$ and using the property that the sum over all Goldberg coefficients of order l is less than 1 ([34] and [35, Lem. A.1]), we have

$$\begin{aligned}\|\text{H.O.T}(5)\|_2 &\leq \sum_{l=5}^{\infty} \|X_k\|_2^{l-1}\|Y_k\|_2 \\ &\leq \sum_{l=5}^{\infty}(1+|\tau|)^{l-1}\|L_k\|_2^{l-1}\left(\kappa\|C_k\|_2 + \frac{\eta}{\xi}\|\text{H.O.T}_A(5)\|_2 + \Psi\right) \\ &\leq \left(\kappa\|C_k\|_2 + \frac{\eta}{\xi}\|\text{H.O.T}(5)\|_2 + \Psi\right)\sum_{l=5}^{\infty}(1+|\tau|)^{l-1}\delta^{l-1} \\ &= \alpha\kappa\|C_k\|_2 + \frac{\eta\alpha}{\xi}\|\text{H.O.T}(5)\|_2 + \Psi,\end{aligned}$$

where $\alpha := \frac{\delta^4(1+|\tau|)^4}{1-\delta(1+|\tau|)}$ and where the condition $\delta(1+|\tau|) < 1$ for the convergence of the series was ensured by Condition 4.4. Also by Condition 4.4, we have $1 - \frac{\eta\alpha}{\xi} > 0$, leading to

$$\|\text{H.O.T}(5)\|_2 \leq \frac{\kappa\alpha}{1-\frac{\eta\alpha}{\xi}}\|C_k\|_2 + \Psi \leq \frac{\kappa\alpha}{1-\frac{\eta\alpha}{\xi}}\|C_k\|_2 + \mathcal{O}(\|C_k\|_2^2). \tag{4.20}$$

Inserting (4.20) in Ψ shows that $\Psi \in \mathcal{O}(\|C_k\|_2^2)$. Finally, we have

$$\begin{aligned}\|C_{k+1}\|_2 &\leq \left\|\frac{1}{6}B_k\Theta_k B_k^T\right\|_2 + \|\text{H.O.T}_C(5)\|_2 + \mathcal{O}(\|C_k\|_2^2) \\ &\leq \frac{\eta\delta^4}{6\xi(6-\delta^2)}\|C_k\|_2 + \left(1+\frac{\eta}{\xi}\right)\|\text{H.O.T}(5)\|_2 + \mathcal{O}(\|C_k\|_2^2) \\ &\leq \left(\frac{\eta\delta^4}{6\xi(6-\delta^2)} + \left(1+\frac{\eta}{\xi}\right)\frac{\kappa\alpha}{1-\frac{\eta\alpha}{\xi}}\right)\|C_k\|_2 + \mathcal{O}(\|C_k\|_2^2).\end{aligned}$$

This concludes the proof. ∎

As seen above, the proof of Theorem 4.3 requires some technical conditions. These conditions are gathered in Condition 4.4 to be seen at a glance. Fig. 2 illustrates the admissible set of pairs (β, δ) satisfying Condition 4.4. Since all inequalities are strict, the set is an open set. In particular, notice that the largest feasible δ occurs at $\beta = \frac{1}{2}$ ($\tau = \eta = \zeta = 0,\ \xi = 1,\ \alpha = \frac{\delta^4}{1-\delta},\ \kappa = \frac{6}{6-\delta^2}$), which corresponds to the canonical metric. In Condition 4.4, we have by definition $\xi \leq 2\beta - \eta \leq 2\beta - |\tau| = 2\beta - |1-2\beta|$. When $\beta \leq \frac{1}{4}$, $2\beta - |1-2\beta| \leq 0$ and the condition $\xi > 0$ can never be satisfied.

Condition 4.4. Given $\beta, \delta > 0$, δ is admissible for Theorem 4.3 if it satisfies

$$\begin{aligned}&\delta(1+|\tau|) < 1, \qquad 1 - \frac{\eta\alpha}{\xi} > 0, \qquad 2\delta|\tau| < 1, \\ &\xi := 2\beta - \eta\left(1 + 2\beta\delta + \frac{4\beta^2}{3}\delta^2 + \frac{\delta^2}{6} + \frac{\delta^3}{6-\delta^2}\right) > 0, \\ &\left(\frac{\eta\delta^4}{6\xi(6-\delta^2)} + \left(1+\frac{\eta}{\xi}\right)\frac{\kappa\alpha}{1-\frac{\eta\alpha}{\xi}}\right) < 1,\end{aligned}$$

where $\tau := 1 - 2\beta,\ \eta := |\tau|(1+\delta|\tau| + \frac{2}{3}\delta^2|\tau|^2 - \delta^2|\tau|^2\log(1-2|\tau|\delta)),\ \alpha := \frac{\delta^4(1+|\tau|)^4}{1-\delta(1+|\tau|)}$ and $\kappa := \max\left(\frac{6}{6-\delta^2}, \frac{\eta\delta^2}{\xi(6-\delta^2)}\right)$.

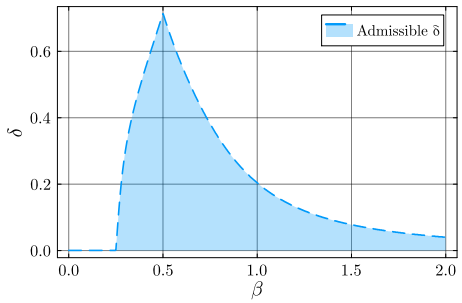

FIG. 2. *Open set of pairs* (β, δ) *satisfying Condition 4.4. Condition 4.4 gathers sufficient conditions to ensure Theorem 4.3 on the convergence of the backward Algorithm 4.1.*

4.4. Local linear convergence of three different forward iterations. We address now the question of the *linear* convergence of Algorithm 4.1 when we use an approximation $\widehat{A}_k \approx A_k$. As before, we examine the convergence rate only for instances for which the algorithm converges. The difference between the forward and the backward case comes from (4.4), thus depending completely on Θ_k. Selecting $\widehat{A}_k$ where $\|\widehat{A}_k\|_2 < \delta$ and replacing A_k by $\widehat{A}_k$ in (4.14) yields

$$\|\Theta_k\|_2 \leq \eta \|\widehat{A}_k - \widehat{A}_{k+1}\|_2. \tag{4.21}$$

If we can find $c > 0$ (typically $c \geq 1$) such that

$$\|\widehat{A}_k - \widehat{A}_{k+1}\|_2 \leq c\|A_k - A_{k+1}\|_2, \tag{4.22}$$

the new forward convergence theorem follows directly by replacing η by $\widehat{\eta} := c\eta$ in Theorem 4.3 and Condition 4.4. In practice, (4.22) means that the approximation $\widehat{A}_k$ converges as fast as A_k does. Obtaining c explicitly in (4.22) happens to be a burden for many approximations $\widehat{A}_k$. However, ensuring the existence of $c \in [0, +\infty)$ is easier. In the next subsections, we propose three linearly convergent ways of choosing $\widehat{A}_k$.

4.4.1. The fixed forward iteration. The computationally cheapest choice for the approximation is $\widehat{A}_k = A_{k-1}$, which we term *fixed forward iteration*. In this case, (4.22) becomes

$$\|A_{k-1} - A_k\|_2 \leq c\|A_k - A_{k+1}\|_2, \tag{4.23}$$

Equation (4.23) is reversed when compared to the definition of linear convergence. Indeed, linear convergence is characterized by the relation $\|A_k - A_{k+1}\|_2 \leq r\|A_{k-1} - A_k\|_2$ for some $r \in (0, 1)$. We will show that (4.23) holds for $k \in \mathbb{N}$ with the constant

$$\overline{c} := \max\left(\sup_{k\in\mathbb{N}} \frac{\|A_{k-1} - A_k\|_2}{\|A_k - A_{k+1}\|_2}, \frac{\|\widehat{A}_0 - A_0\|_2}{\|A_0 - A_1\|_2}\right) \in [0, +\infty). \tag{4.24}$$

The second argument of the max is only introduced for subsection 4.4.3. Assume there is no static iteration before convergence, i.e., no k such that $A_k = A_{k+1}$. The property $\overline{c} \geq 0$ is a direct consequence of its definition. The fact that $\overline{c} < +\infty$ holds if and only if there is no super-linear convergence of the sequence $\{\|A_k - A_{k+1}\|_2\}_{k\in\mathbb{N}}$ to 0. In case this sequence converges super-linearly to 0, the sequence $\{\|\Theta_k\|_2\}_{k\in\mathbb{N}}$ does too, by (4.21). Then, Θ_k can simply be ignored from our analysis and we are done.

We can thus focus on the case of no super-linear convergence of $\{\|A_k - A_{k+1}\|_2\}_{k\in\mathbb{N}}$, which ensures the existence of $\overline{c} \in [0, +\infty)$. The caveat of the fixed forward iteration is that we have no control on $\overline{c}$ — that is, on the convergence rate. It is fixed by the inputs of Algorithm 4.1 and we can only observe the consequences. This motivates looking for alternative forward iterations. In the next subsection, we introduce such an alternative, termed *pseudo-backward*.

Remark 4.5. For all types of forward iterations, the first estimate $\widehat{A}_0$ must be chosen by a given rule. We propose to compute it using a BCH series expansion in subsection 4.5.

4.4.2. The pseudo-backward iteration ($c = 1 + \nu + \nu\overline{c}_\nu$ in (4.22)). By a pseudo-backward iteration, we mean an iteration scheme where $\widehat{A}_k$ satifies $\|\widehat{A}_k - A_k\|_2 \leq \nu\|A_k - A_{k-1}\|_2$ with $\nu \in [0, 1]$. Said otherwise, $\widehat{A}_k$ ensures a sufficient improvement in each iteration compared to the fixed forward approximation A_{k-1}. This intentionally general definition matches very well all cases where we approximately obtain the backward iterate using an iterative method, for instance Algorithm 4.2. This intermediate type of iteration is designed to converge faster than the fixed forward case without requiring the exact computation of the backward iterate A_k. Notice that $\nu = 1$ corresponds to a fixed forward iteration while $\nu = 0$ corresponds to a backward iteration. This new iteration leads to

$$\begin{aligned}\|\widehat{A}_k - \widehat{A}_{k+1}\|_2 &= \|\widehat{A}_k - \widehat{A}_{k+1} + A_k - A_{k+1} - A_k + A_{k+1}\|_2 \\ &\leq \|\widehat{A}_k - A_k\|_2 + \|\widehat{A}_{k+1} - A_{k+1}\|_2 + \|A_{k+1} - A_k\|_2 \\ &\leq \nu\|A_k - A_{k-1}\|_2 + \nu\|A_{k+1} - A_k\|_2 + \|A_{k+1} - A_k\|_2 \\ &\leq (1 + \nu + \nu\overline{c}_\nu)\|A_{k+1} - A_k\|_2,\end{aligned} \tag{4.25}$$

where $\overline{c}_\nu$ is defined as in (4.24), but is affected by ν. The goal here is to enhance the convergence rate by moderating the constant $\overline{c}_\nu$ by the adaptive factor ν such that $1 + \nu + \nu\overline{c}_\nu \ll \overline{c}$. We design a pseudo-backward algorithm in subsection 5.2.

4.4.3. The accelerated forward iteration ($c = \frac{(1+|h|)\overline{c}_h}{1-\overline{c}_h|h|}$ in (4.22)). The goal of this third and last type of iteration is to build an improved approximation $\widehat{A}_k$ using the information encoded in the previous-stage gap $A_{k-1} - \widehat{A}_{k-1}$. We define the accelerated forward iteration by

$$\widehat{A}_k = A_{k-1} + hQ_{k-1}(A_{k-1} - \widehat{A}_{k-1})Q_{k-1}^T, \tag{4.26}$$

where $h \in \mathbb{R}$ and $Q_{k-1} \in \mathrm{O}(p)$. The step size h controls the importance given to the momentum while the matrix Q_{k-1} performs a change of basis that aims at accounting that A_{k-1} and $A_{k-1} - \widehat{A}_{k-1}$ are expressed in different bases. A concrete example is given at the end of this section. This new approximant $\widehat{A}_k$ should be consistent, i.e., the sequence $\{\|\widehat{A}_k - A_k\|_2\}_{k\in\mathbb{N}}$ should converge linearly to 0 when $\{\|A_k - A_{k-1}\|_2\}_{k\in\mathbb{N}}$ does. This was true by definition in the two previous cases. For the accelerated forward iteration, it follows from

$$\begin{aligned}\|\widehat{A}_k - A_k\|_2 &\leq \|A_k - A_{k-1}\|_2 + |h|\|Q_{k-1}(\widehat{A}_{k-1} - A_{k-1})Q_{k-1}^T\|_2 \\ &= \|A_k - A_{k-1}\|_2 + |h|\|\widehat{A}_{k-1} - A_{k-1}\|_2 \\ &\leq \sum_{l=0}^{k-1} |h|^l\|A_{k-l} - A_{k-1-l}\|_2 + |h|^k\|\widehat{A}_0 - A_0\|_2 \qquad \text{(by recursion)}\end{aligned}$$

$$\leq \|A_k - A_{k-1}\|_2 \sum_{l=0}^{k} \overline{c}_h^l |h|^l \qquad \text{(by (4.23) and (4.24))}$$

$$\leq \frac{1}{1-\overline{c}_h|h|}\|A_k - A_{k-1}\|_2 \qquad \text{(if } \overline{c}_h|h| < 1\text{)}$$

Here, $\overline{c}_h$ is defined as in (4.24) but depends on h. Therefore, the accelerated forward iteration is consistent for h small enough. In addition to consistency, $\{\widehat{A}_k\}_{k\in\mathbb{N}}$ should be convergent by satisfying (4.22). This holds since

$$\begin{aligned}
\|\widehat{A}_{k+1} - \widehat{A}_k\|_2 &= \|A_k - \widehat{A}_k - hQ_k(\widehat{A}_k - A_k)Q_k^T\|_2 \\
&\leq (1+|h|)\|\widehat{A}_k - A_k\|_2 \\
&\leq \frac{1+|h|}{1-\overline{c}_h|h|}\|A_k - A_{k-1}\|_2 \\
&\leq \frac{(1+|h|)\overline{c}_h}{1-\overline{c}_h|h|}\|A_{k+1} - A_k\|_2 \qquad \text{(if } \overline{c}_h|h| < 1\text{)}
\end{aligned}$$

Notice that for $h = 0$, we retrieve $c = \overline{c}$, the fixed forward iteration. Since $\overline{c}_h$ continuously depends on h and since $\overline{c}_h h = 0$ for $h = 0$, there is an open neighbourhood of $h = 0$ for which $\overline{c}_h|h| < 1$. The constant $c = \frac{(1+|h|)\overline{c}_h}{1-\overline{c}_h|h|}$ is worse than $c = \overline{c}$, obtained for the fixed forward iteration. This is because a bad step size h, notably a wrong sign of h, can make Algorithm 4.1 converge slower.

Choosing Q_k and h. We propose an accelerated forward iteration achieving fast numerical convergence, as will be confirmed in subsection 5.1. To this end, let $\tau := 1 - 2\beta$ and set $h := -\tau$, $Q_k := \exp(-\tau A_k)$ so that (4.26) becomes

$$\widehat{A}_{k+1} = A_k - \tau \exp(-\tau A_k)(A_k - \widehat{A}_k)\exp(\tau A_k). \tag{4.27}$$

This choice is based on a geometric motivation, illustrated in Fig. 3. By extrapolating the shift of the approximation from $\widehat{A}_k$ to A_k, we update the current iterate A_k using the parallel transport at zero time of a vector field $\Delta_k(t)$ along the geodesic $\gamma : [0,1] \mapsto \mathrm{SO}(p) : t \to \exp(-\tau A_k t)$, see, e.g., [13, p. 9]. Recall that any such vector field along γ can be parameterized by $\Delta_k(t) = \exp(-\tau A_k t)W_k(t)$ where $W_k(t) \in \mathrm{Skew}(p)$. Here, we choose the momentum $\Delta_k(t)$ at unit time by

$$\Delta_k(1) := \exp(-\tau A_k)W_k(1) := \exp(-\tau A_k)\log\left(\exp(-\tau A_k)\exp(\tau\widehat{A}_k)\right). \tag{4.28}$$

However, since $\Delta_k(1) \in T_{\exp(-\tau A_k)}\mathrm{SO}(p)$ and $A_k \in T_{I_p}\mathrm{SO}(p)$, these two quantities cannot be directly summed: we must first compute the parallel transport of $\Delta_k(1)$ to zero time, $\Delta_k(0) \in T_{I_p}\mathrm{SO}(p)$, before updating $\widehat{A}_{k+1} = A_k + \alpha_k W_k(0)$ for some step size $\alpha_k \in \mathbb{R}$. It follows from [13, Eq. 2.18] that the parallel transport is given by

$$\Delta_k(1) = I_p \exp\left(-\frac{\tau A_k}{2}\right) W_k(0) \exp\left(-\frac{\tau A_k}{2}\right). \tag{4.29}$$

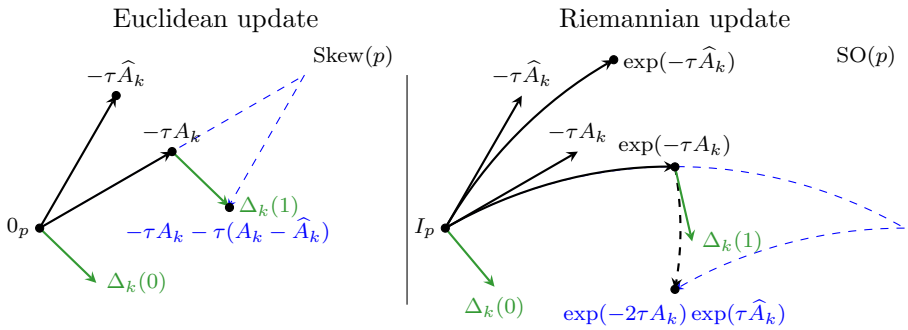

FIG. 3. *An artist view of the geometric construction of the momentum Δ_k. On the left, the figure shows a construction that is equivalent to (4.28) in the context of a flat space, e.g.,* $\mathrm{Skew}(p)$ *equipped with the standard inner product $\langle\cdot,\cdot\rangle_{\mathrm{F}}$. On the right, the figure illustrates the analogous construction of Δ_k in the context of the Riemannian manifold* $\mathrm{SO}(p)$ *viewed as a subset of* $(\mathrm{St}(p,p),\langle\cdot,\cdot\rangle_{\beta=1})$.

By combining (4.28) and (4.29), we have $W_k(0) = \exp\left(-\frac{\tau A_k}{2}\right) W_k(1) \exp\left(\frac{\tau A_k}{2}\right)$ and, therefore, the update is

$$\begin{aligned}\widehat{A}_{k+1} =& A_k + \alpha_k \exp\left(-\frac{\tau A_k}{2}\right) \log\left[\exp(-\tau A_k)\exp(\tau\widehat{A}_k)\right] \exp\left(\frac{\tau A_k}{2}\right)\\ =& A_k + \alpha_k \exp\left(-\tau A_k\left(\frac{1}{2}+\omega\right)\right)\cdot\\ & \log\left[\exp\left(-\tau A_k(1-\omega)\right)\exp(\tau\widehat{A}_k)\exp\left(-\omega\tau A_k\right)\right]\exp\left(\tau A_k\left(\frac{1}{2}+\omega\right)\right),\end{aligned} \tag{4.30}$$

where $\omega \in \mathbb{R}$ is a free parameter. The purpose of introducing ω is that the numerical scheme of (4.27) arises by choosing $\alpha_k = 1$ and by linearizing the costly matrix logarithm of (4.30) using the BCH formula by

$$\begin{aligned}&\log\left[\exp\left(-\tau A_k(1-\omega)\right)\exp(\tau\widehat{A}_k)\exp\left(-\omega\tau A_k\right)\right] =\\ &\qquad -\tau(A_k-\widehat{A}_k) + \frac{\tau^2(1-2\omega)}{2}[\widehat{A}_k, A_k] + \text{ H.O.T}(3).\end{aligned} \tag{4.31}$$

Equation (4.31) shows that choosing $\omega = \frac{1}{2}$ yields a linearization of higher-order accuracy. In addition to be cheaper than (4.30), experiments suggest that (4.27) achieves the same performance as (4.30) in terms of number of iterations of Algorithm 4.1. A likely explanation is that the higher accuracy of (4.30) is lost throughout the iteration loop of Algorithm 4.1. Current attempts to demonstrate theoretically the effectiveness of this choice of accelerated forward iteration remained unfruitful due to the several layers of nonlinearities that are involved.

4.5. An efficient start point for the forward iteration. A good initial guess for $\widehat{A}_0$ is important for the fast convergence of Algorithm 4.1. Starting from (4.2) and defining $\log(V_0) := \begin{bmatrix} E & -F^T \\ F & G \end{bmatrix} \in \mathrm{Skew}(2p)$, the BCH series expansion of the $p \times p$ top-left block of (4.2) implies that

$$2\beta(A_0 - \widehat{A}_0) = E - \widehat{A}_0 + \frac{\tau}{12}(F^T F\widehat{A}_0 + \widehat{A}_0 F^T F) - \frac{\tau}{2}[E, \widehat{A}_0] + \text{H.O.T.} \tag{4.32}$$

To obtain a good approximation $\widehat{A}_0 \approx A_0$, we choose $\widehat{A}_0$ in such a way that the three first right terms of (4.32) cancel out. Hence, $\widehat{A}_0$ must be a solution of the Sylvester

equation

$$S\widehat{A}_0 + \widehat{A}_0 S = E \text{ where } S := \frac{I_p}{2} - \frac{\tau}{12} F^T F. \tag{4.33}$$

If $\beta \geq \frac{1}{2}$ ($\tau \leq 0$), the smallest eigenvalue of S is bounded from below by $\frac{1}{2}$. It follows from [6, Thm. VII.2.12], that $\|\widehat{A}_0\|_2 \leq \|E\|_2 \leq \delta$. If $0 < \beta < \frac{1}{2}$ ($\tau > 0$), the smallest eigenvalue of S is bounded from below by $\frac{1}{2} - \frac{\tau}{12}\delta^2$, thus $\|\widehat{A}_0\|_2 \leq \frac{6}{6-\tau\delta^2}\|E\|_2 \leq \frac{6\delta}{6-\tau\delta^2}$. In this paper, all the experiments are performed using this first initial guess $\widehat{A}_0$.

4.6. A quasi-geodesic sub-problem for the backward iteration. We have not yet addressed the question of how to perform a *backward iteration*. Equation (4.1) has to be solved. Following [3], it can be termed "finding a *quasi-geodesic*", stated in Problem 4.6.

Problem 4.6. QUASI-GEODESIC SUB-PROBLEM Given $V \in \mathrm{SO}(2p)$ and $\beta > 0$, find $D,\, C \in \mathrm{Skew}(p),\ B \in \mathbb{R}^{p\times p}$ such that

$$\gamma(t) := \exp\left(t \begin{bmatrix} 2\beta D & -B^T \\ B & C \end{bmatrix}\right) \begin{bmatrix} \exp(t(1-2\beta)D) & 0 \\ 0 & I_p \end{bmatrix} \text{ ends in } \gamma(1) = V. \tag{4.34}$$

Algorithm 4.2 is a method to solve Problem 4.6. It is a simplified version of Algorithm 4.1 where C_k is not constrained to converge to 0 anymore. Starting from an initial guess $\widehat{D}_0$ (A_{k-1} in practice), we successively obtain D_k and update $\widehat{D}_{k+1}$ as the accelerated forward approximation of D_{k+1} for $k = 0, 1, \ldots$. Then, we set $\widehat{A}_k = D_\infty$ and perform the next backward iteration of Algorithm 4.1. Solving Problem 4.6 to ε-precision using Algorithm 4.2 is however not competitive with the previously proposed methods [9, 32, 36]. This is why we investigate a *pseudo-backward* iteration in subsection 5.2 — we only perform a few iterations of Algorithm 4.2 to improve the approximation $\widehat{A}_k$. An alternative method to solve Problem 4.6 based on shooting principle from [9] is proposed in Appendix D. In practice, we observed the better performance of Algorithm 4.2.

Algorithm 4.2 The sub-problem's iterative algorithm

INPUT: Given $V \in \mathrm{SO}(2p)$, $\widehat{D}_0 \in \mathrm{Skew}(p)$, $\beta > 0$ and $\varepsilon > 0$, compute:
Define $\tau = 1 - 2\beta$.
for $k = 0, 1, \ldots$ **do**
  Compute $\begin{bmatrix} 2\beta D_k & -B_k^T \\ B_k & C_k \end{bmatrix} = \log\left(V \begin{bmatrix} \exp(-\tau \widehat{D}_k) & 0 \\ 0 & I_p \end{bmatrix}\right)$.
  if $\|D_k - \widehat{D}_k\|_{\mathrm{F}} < \varepsilon$ **then**
    Break.
  end if
  Set $\widehat{D}_{k+1} = D_k - \tau \exp(-\tau D_k)(D_k - \widehat{D}_k)\exp(\tau D_k)$.
end for
return D_k

5. The performance of forward iterations. This section investigates the numerical convergence of the different forward iterations introduced in subsection 4.4. These variants of Algorithm 4.1 are benchmarked in subsection 6.2.

5.1. The accelerated forward iteration. Subsection 4.4.3 defined the accelerated forward iteration. Fig. 4 quantifies how much this strategy speeds-up Algorithm 4.1 compared to using the fixed forward iteration. First, notice in Fig. 4 that

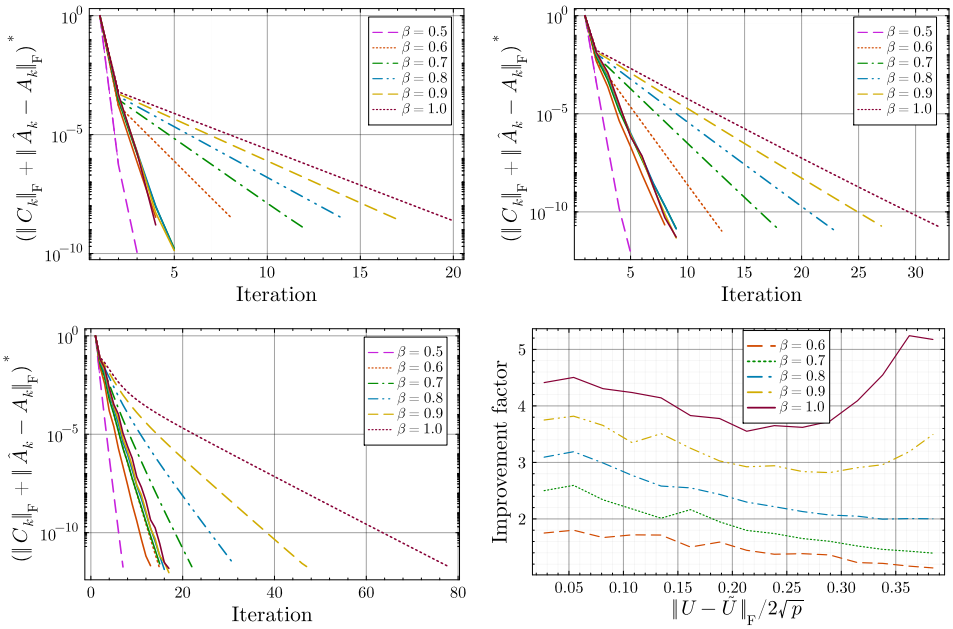

FIG. 4. *Convergence of Algorithm 4.1 with an accelerated forward iteration ($Q_k = \exp((2\beta - 1)A_k)$) from subsection 5.1 (solid lines) and the fixed forward iteration (stylized lines) on* $\mathrm{St}(n = 120, p = 50)$. *The matrices $U, \widetilde{U}$ are randomly generated at Frobenius distance $\|U - \widetilde{U}\|_{\mathrm{F}} \in \{0.03, 0.19, 0.37\} \cdot 2\sqrt{p}$ for respectively the top left, top right and bottom left plots. The stars "$*$" on the y-axes specify that the residuals are normalized by the residual of the first iteration. The bottom right figure shows how the improvement factor (i.e., the ratio between the number of iterations of the fixed forward and the accelerated forward method) varies as the Frobenius distance increases in* $\mathrm{St}(n = 60, p = 30)$.

the convergence rate of the fixed forward iteration is very sensitive to β. In comparison, the accelerated forward method provides a convergence rate that is almost independent of β when it converges. The improvement effect as a function of β and $\|\widetilde{U} - U\|_F$ is summarized on the bottom right plot of Fig. 4. The fast increase of the improvement factor for $\beta = 1$ when the Frobenius distance gets larger is due to the increased convergence radius of the accelerated forward iteration, while the fixed forward iteration gets close to divergence.

5.2. The pseudo-backward iteration compared to the fixed forward iteration. Pseudo-backward iterations only perform a few sub-iterations of Algorithm 4.2 with $\widehat{D}_0 := A_{k-1}$ to improve the quality of the estimation $\widehat{A}_k$ such that $\|\widehat{A}_k - A_k\|_F \leq \nu \|A_{k-1} - A_k\|_F$ with $\nu < 1$. The left plot of Fig. 5 exemplifies that performing two sub-iterations can already provide $\nu \approx 0.1$. Combine this result with a second observation: Algorithm 4.1 will produce a new iterate A_{k+1} anyway so that it is not worth the computational effort to obtain $\widehat{A}_k$ with $\|\widehat{A}_k - A_k\|_{\mathrm{F}} \ll \|A_{k+1} - A_k\|_{\mathrm{F}}$. Hence, the first iterations of Algorithm 4.2 speed-up Algorithm 4.1 but the last ones are a waste of resource. Fig. 5 investigates the optimal number of sub-iterations to perform. The further β is from $\frac{1}{2}$, the more the sub-iterations are improving the performance. For $\beta \in [0.7, 1]$, the optimal number of sub-iterations settles at 2.

Fig. 6 illustrates that the more sub-iterations of Algorithm 4.2 are performed, the fewer iterations of Algorithm 4.1 are needed. For these experiments, 2 sub-iterations are enough to reach the performance of the backward iteration.

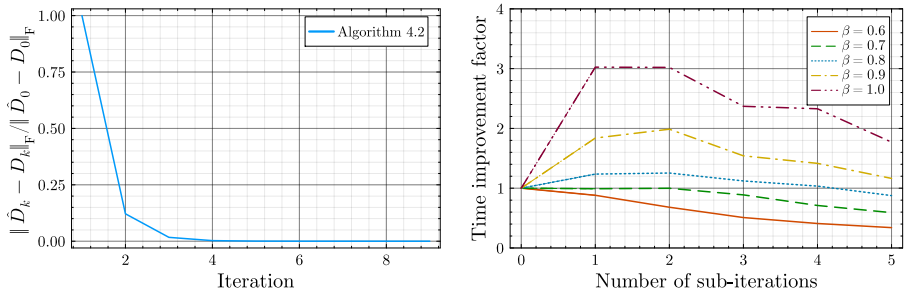

FIG. 5. *On the left, the evolution of the residual* $\|D_k - \widehat{D}_k\|_{\mathrm{F}}$ *of Algorithm 4.2 for a random matrix* $V \in \mathrm{SO}(80)$ *with* $\|V - I_{80}\|_F = 0.36 \cdot 2\sqrt{40}$. *On the right, the* time improvement factor *is the ratio between the running time of the fixed forward and the pseudo-backward method when* β *and the number of sub-iterations vary. The stopping criterion is set to* $(\|C_k\|_F + \|\widehat{A}_k - A_k\|_F)^* < 10^{-12}$. *The experiment is performed on* $\mathrm{St}(n = 80, p = 40)$ *at distance* $\|U - \widetilde{U}\|_{\mathrm{F}} = (0.36 \pm 0.01) \cdot 2\sqrt{p}$.

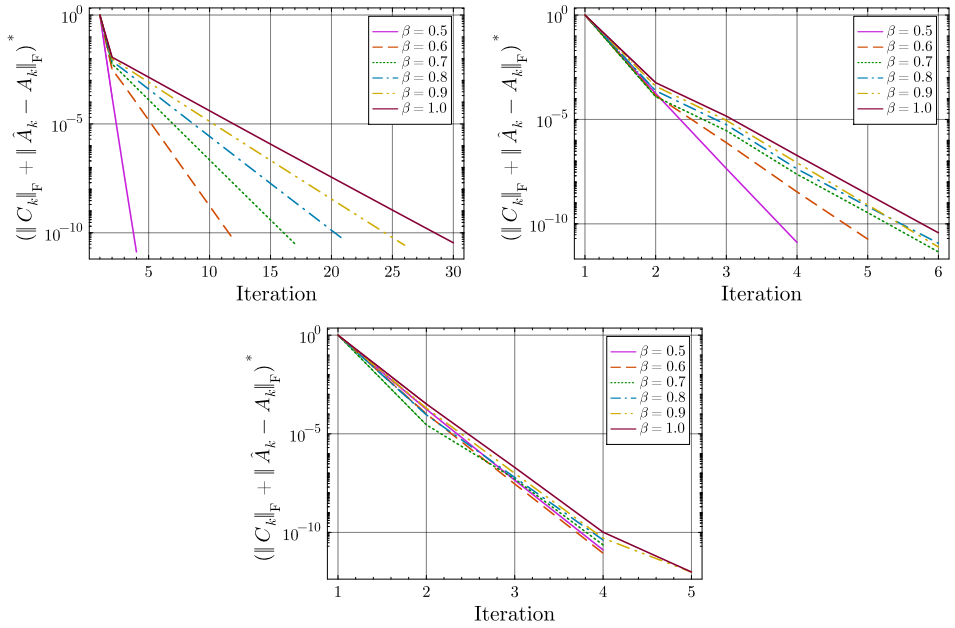

FIG. 6. *Evolution of the residuals* $\|C_k\|_{\mathrm{F}} + \|A_k - \widehat{A}_k\|_{\mathrm{F}}$ *for the fixed forward iteration (top left), pseudo-backward iteration with 1 sub-iterations (top right) and 2 sub-iterations (bottom) for a random experiment on* $\mathrm{St}(n = 80, p = 30)$ *with* $\|U - \widetilde{U}\|_{\mathrm{F}} = 0.16 \cdot 2\sqrt{p} \approx 1.75$. *The star "* $*$*" indicates that the residuals are normalized by the residual of the first iteration. For all plots, the curve for* $\beta = 0.5$ *(solid purple line) is the same since the all iterations reduce to the same algorithm for the canonical metric.*

6. Performance analysis. We investigate the performance of Algorithm 4.1 through two questions: how often and how fast does it converge? To answer the first one, we estimate a probabilistic convergence radius— that is, we find the distance $\|U - \widetilde{U}\|_{\mathrm{F}}$ such that Algorithm 4.1 converges with probability close to 1, say 0.99. For the second question, we compare the running times of known methods, carefully implemented to extract the best performance for each of them.

6.1. Probabilistic radius of convergence. From randomized numerical experiments, we fit a logistic model $m_\theta(x) \approx P(\text{Alg. 4.1 converged} \mid \|\widetilde{U} - U\|_{\mathrm{F}} = x)$ where θ is chosen as the maximum likelihood estimator. The details and motivation of this fitting are described in Appendix E. The left plot in Fig. 7 displays the fitting

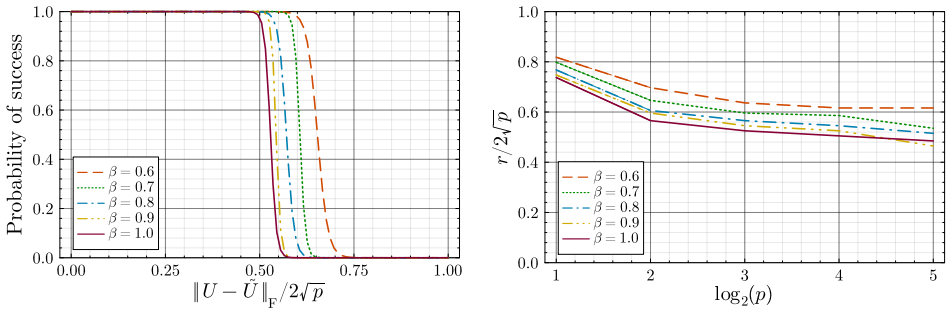

FIG. 7. *On the left, the logistic regression models on 1000 random samples on* $\mathrm{St}(32, 16)$. *The models are trained until* $\|\nabla f(\theta)\|_F < 10^{-8}$ *(see Appendix E). The logistic model estimates the probability of success of Algorithm 4.1 (implemented with 2 pseudo-backward sub-iterations). Respectively for* β *from* 0.6 *to* 1, *the R2 factors of the fitting are* $\{0.85, 0.97, 0.96, 0.98, 0.97\}$, *confirming the goodness of fit. On the right, the evolution of the radius of convergence as* p *varies on* $\mathrm{St}(2p, p)$.

models on $\mathrm{St}(32, 16)$. This fit provides a probabilistic radius of convergence r by taking $r := \max\{x \in \mathbb{R} \mid m_\theta(x) \geq 0.99\}$, i.e., the probability of convergence is higher than 99% if $\|U - \widetilde{U}\|_F \leq r$. The right plot in Fig. 7 shows that convergence is most challenging under the Euclidean metric ($\beta = 1$). In this case, Algorithm 4.1 converges with high probability if the distance of the inputs is $\|U - \widetilde{U}\|_F < 0.4 \cdot 2\sqrt{p}$.

Remark 6.1. When $U, \widetilde{U}$ are not in the convergence radius of Algorithm 4.1, it is still possible to use it as a subroutine for a globally convergent method, e.g., the *leapfrog method* of [27]. This method has already been considered on the Stiefel manifold [32, 33]. It was then combined with a shooting method. However, being a global method, the leapfrog method needs not to be used inside the convergence radius of Algorithm 4.1.

6.2. Benchmark. We benchmark the different versions of Algorithm 4.1 with the p-shooting method [36, Algorithm 2]. The benchmarking is performed on random matrices $U_i, \widetilde{U}_i \in \mathrm{St}(n, p)$ for $i = 1, ..., N$ generated at fixed Frobenius distance. In view of the homogeneity of $\mathrm{St}(n, p)$, the U_i's can be chosen arbitrarily w.l.o.g, here by applying a Gram-Schmidt process on a random matrix. The $\widetilde{U}_i$'s are built within the convergence radius as follows. For $A_i \in \mathrm{Skew}(n)$ filled with i.i.d normally distributed entries $\sim \mathcal{N}(0, \delta)$, we build $\widetilde{U}_i = [U_i \; U_{i,\perp}] \exp(A_i) I_{n \times p}$. Because of a "Central Limit Theorem effect", if δ is fixed and n, p are large, the value of $\frac{\|U_i - \widetilde{U}_i\|_F^2}{4p} \in [0, 1]$ converges to an expected value. Our experiments compare

1. Algorithm 4.1 using fixed forward iterations.
2. Algorithm 4.1 using pseudo-backward iterations with 1 and 2 sub-iterations of Algorithm 4.2.
3. Algorithm 4.1 using accelerated forward iterations.
4. The p-shooting method [36, Algorithm 2] with 3 and 5 discretization points, implemented as in RiemannStiefelLog.

We consider two sizes of Stiefel manifolds, $\mathrm{St}(80, 20)$ and $\mathrm{St}(100, 50)$. We also consider two distances between the samples, expressed in percentage of the Frobenius diameter of $\mathrm{St}(n, p)$ $(= 2\sqrt{p})$. We consider 15% and 32%. Given a method M that ran T times with CPU times $t_{M,j}(U_i, \widetilde{U}_i)$ for $j = 1, \ldots, T$, the running time t_M of the method M is computed as follows:

$$t_M = \frac{1}{N} \sum_{i=1}^{N} \min_{j=1,\ldots,T} t_{M,j}(U_i, \widetilde{U}_i). \tag{6.1}$$

St(80, 20), $\|U - \widetilde{U}\|_F = (0.15 \pm 0.01) \cdot 2\sqrt{p}$								
β	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Scale	$\cdot 10^{-2}$ sec.							
Fixed Forward	2.8	1.0	0.31	0.84	1.1	1.35	1.6	1.8
Pseudo-Backward 1 it.	1.5	0.93	0.31	0.76	0.86	0.92	0.88	0.89
Pseudo-Backward 2 it.	1.35	0.85	0.31	0.85	0.84	0.85	0.86	0.93
Acc. Forward	1.30	0.87	0.31	0.79	0.81	0.85	0.81	0.80
p-shooting 3 pt.	2.3	2.1	1.8	1.8	1.5	1.3	15(✗)	0.92
p-shooting 5 pt.	2.9	2.7	2.2	2.2	2.0	1.8	1.4	1.2

St(80, 20), $\|U - \widetilde{U}\|_F = (0.32 \pm 0.01) \cdot 2\sqrt{p}$								
β	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Scale	$\cdot 10^{-2}$ sec.							
Fixed Forward	3.3	1.2	0.44	0.95	1.2	1.6	1.9	2.2
Pseudo-Backward 1 it.	2.7	1.3	0.45	1.1	1.2	1.3	1.5	1.6
Pseudo-Backward 2 it.	2.6	1.4	0.42	1.4	1.4	1.3	1.5	1.6
Acc. Forward	2.7	1.3	0.45	1.1	1.3	1.3	1.3	1.3
p-shooting 3 pt.	4.3	3.8	2.8	2.8	2.4	17(✗)	1.7	1.6
p-shooting 5 pt.	5.3	4.6	3.6	3.4	2.9	2.6	2.3	1.8

St(100, 50), $\|U - \widetilde{U}\|_F = (0.32 \pm 0.01) \cdot 2\sqrt{p}$								
β	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Scale	$\cdot 10^{-1}$ sec.							
Fixed Forward	5.2	1.8	0.71	1.5	2.1	3.0	4.3	7.4
Pseudo-Backward 1 it.	5.6	1.6	0.72	1.4	1.7	1.9	2.0	2.1
Pseudo-Backward 2 it.	5.8	1.8	0.71	1.5	1.7	1.7	1.9	1.9
Acc. Forward	8.1	1.9	0.72	1.4	1.7	1.7	1.6	1.7
p-shooting 3 pt.	38(✗)	5.1	12(✗)	3.1	2.5	33(✗)	1.6	22(✗)
p-shooting 5 pt.	18(✗)	5.5	3.5	3.3	2.6	2.1	1.9	1.6

TABLE 1

Benchmark of Algorithm 4.1 in its different versions (fixed forward, pseudo-backward with 1 or 2 sub-iterations and accelerated forward). It is compared to the p-shooting method [36, Algorithm 2] with 3 and 5 points (starting and ending point included). With only 3 points, the appearance of failures (✗) indicates that more points must be considered for robustness. For each value of β, the cell of the best performing method is highlighted.

The "min" in (6.1) is more standard than the average for benchmarking, as implemented for the `@btime` macro in `Julia` [5]. For Table 1, we take $N = T = 10$. The experiments from Table 1 demonstrate the effectiveness of Algorithm 4.1 compared to the shooting method for $\beta \in [0.3, 1]$. The shooting method is only competitive with Algorithm 4.1 when $\beta = 1$, i.e., in the case of the Euclidean metric. In particular the accelerated forward iteration appears to be the best choice since it is very close in performance to the other types of iterations in case $\beta \leq \frac{1}{2}$ and outperforms them otherwise.

7. Conclusion. In this paper, we have proposed an alternative to the shooting method to compute a geodesic between any two points on the Stiefel manifold. The method generalizes the approach of [35] to the complete family of metrics introduced in [18]. Our analysis included theoretical guarantees and numerical experiments. Future work may include the design of a more robust initialisation, increasing significantly the current restrictive radius of convergence. A more detailed understanding of the effectiveness of the accelerated forward iteration should also be carried out.

8. Acknowledgments. The authors would like to thank Pierre-Antoine Absil for his invaluable guidance during the preparation of this manuscript and anonymous referees for their constructive comments and valuable suggestions, which have significantly improved the quality of this work.

Appendix A. Rank of logarithm. Given $U, \widetilde{U} \in \mathrm{St}(n,p)$, we investigate the cases where $\Delta \in \mathrm{Log}_{\beta,U}(\widetilde{U})$ and

$$\mathrm{rank}((I-UU^T)\Delta) > \mathrm{rank}((I-UU^T)\widetilde{U}). \tag{A.1}$$

An example where (A.1) holds is for antipodal points ($\widetilde{U} = -U$) on the hypersphere $\mathrm{St}(n,1)$. We show that (A.1) can only happen when $\widetilde{U}$ belongs to the cut locus of U, which is a zero-measure subset of $\mathrm{St}(n,p)$ [30, Lemma 4.4].

PROPOSITION A.1. *Assume $U, \widetilde{U} \in \mathrm{St}(n,p)$ and (A.1) holds. Then $\widetilde{U}$ belongs to the cut locus of U.*

Proof. Assume $\Delta \in \mathrm{Log}_{\beta,U}(\widetilde{U})$. By definition, we can always write $\Delta = UA + \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$ where $\mathrm{col}([U\ Q_1]) = \mathrm{col}([U\ \widetilde{U}])$ and $Q_2^T U = Q_2^T \widetilde{U} = 0$. By the assumption (A.1), we have $Q_2, B_2 \neq 0$. Let $Q_2 \in \mathrm{St}(n,q)$ with $q \geq 1$. Then, we have

$$\begin{aligned}
&\widetilde{U} = \mathrm{Exp}_{\beta,U}(\Delta) \\
\iff &\begin{bmatrix} U^T\widetilde{U} \\ Q_1^T\widetilde{U} \\ 0 \end{bmatrix} = \exp \begin{bmatrix} 2\beta A & -B_1^T & -B_2^T \\ B_1 & & \\ B_2 & & 0_{n-p} \end{bmatrix} I_{n\times p} \exp(\tau A) \\
\iff &\begin{bmatrix} U^T\widetilde{U} \\ Q_1^T\widetilde{U} \\ 0 \end{bmatrix} = \exp \begin{bmatrix} 2\beta A & -B_1^T & -(RB_2)^T \\ B_1 & & \\ RB_2 & & 0_{n-p} \end{bmatrix} I_{n\times p} \exp(\tau A) \text{ for all } R \in \mathrm{O}(q).
\end{aligned}$$

Therefore, for all $R \in \mathrm{O}(q)$ and $\widetilde{\Delta} := UA + \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} B_1 \\ RB_2 \end{bmatrix}$, $\mathrm{Exp}_{\beta,U}(\widetilde{\Delta}) = \widetilde{U}$ and $\|\widetilde{\Delta}\|_\beta = \|\Delta\|_\beta$ yield $\widetilde{\Delta} \in \mathrm{Log}_{\beta,U}(\widetilde{U})$. In conclusion, there is more than one minimal geodesic from U to $\widetilde{U}$ and $\widetilde{U}$ is on the cut locus of U by [30, Proposition 4.1]. ∎

Appendix B. Initialization of V_0. Both Algorithm 3.1 and Algorithm 4.1 ask for an initialization $V_0 = \left[\begin{smallmatrix} M & O_0 \\ N & P_0 \end{smallmatrix}\right] \in \mathrm{SO}(2p)$. $M = U^T\widetilde{U}$ is fixed by the problem. However, N, O_0, P_0 and Q have degrees of freedom to obtain the best performance. Let us recall their meaning. From Theorem 2.1, we know that for $n \leq 2p$, the geodesic can be considered as a curve evolving in $\mathrm{St}(n,2p)$, from which we only retain the p first columns. In $\mathrm{St}(n,2p)$, the geodesic starts at $[U\ Q]$ and ends at $[\widetilde{U}\ \widetilde{Q}]$. The underlying goal of Algorithm 3.1 and Algorithm 4.1 is to obtain $Q, \widetilde{Q} \in \mathrm{St}(n,p)$ minimizing the distance from $[U\ Q]$ to $[\widetilde{U}\ \widetilde{Q}]$. From this framework, V_0 can be interpreted as

$$V_0 := \begin{bmatrix} U^T\widetilde{U} & U^T\widetilde{Q} \\ Q^T\widetilde{U} & Q^T\widetilde{Q} \end{bmatrix} \in \mathrm{SO}(2p). \tag{B.1}$$

A heuristic initialization is finding $Q, \widetilde{Q}$ minimizing the geodesic distance $d(I, V_0)$ in $\mathrm{SO}(2p)$. It is known that $d(I,V_0) \propto \|\log(V_0)\|_F$ (see, e.g., [13]). Since $U, \widetilde{U}$ are fixed, an easier problem is to solve $\min_{Q,\widetilde{Q}} \|I - Q^T\widetilde{Q}\|_F$. Since the column spaces of $Q, \widetilde{Q}$ are known, this second problem is an Orthogonal Procrustes problem, solved by the SVD. It admits an infinite set of solutions, given by all orthogonal similarity transformations of $Q^T\widetilde{Q}$. Let us build these solutions. If we take $\widehat{Q}\widehat{N} = (I - UU^T)\widetilde{U}$ (with $\widehat{Q} \in \mathrm{St}(n,p)$ and $\widehat{Q}^T U = 0$), we can start with any $\left[\begin{smallmatrix} M & \widehat{O}_0 \\ \widehat{N} & \widehat{P}_0 \end{smallmatrix}\right] \in \mathrm{SO}(2p)$. The idea from [35] is to compute $\widehat{P}_0 = R\Sigma\widetilde{R}^T$, a singular value decomposition.

- Method from [35]: take $Q := \widehat{Q}$, $N := \widehat{N}$, $O_0 := \widehat{O}_0\widetilde{R}R^T$, $P_0 = R\Sigma R^T$.
- Our method: take $Q := \widehat{Q}R$, $N = R^T\widehat{N}$, $O_0 = \widehat{O}_0\widetilde{R}$, $P_0 = \Sigma$.

Our method corresponds to a similarity transformation of the method from [35] by $\left[\begin{smallmatrix} I & 0 \\ 0 & R \end{smallmatrix}\right]$. It offers a diagonal matrix P_0.

Appendix C. Intermediate results for subsection 4.3. This appendix gathers intermediate results in the proof of Theorem 4.3. It allows to keep a clearer narration in subsection 4.3. First, Lemma C.1 allows to relate the norm of two matrices appearing in Theorem 4.3.

Lemma C.1. Let $L_k := \begin{bmatrix} A_k & -B_k^T \\ B_k & C_k \end{bmatrix}$ and $X_k := \begin{bmatrix} 2\beta A_k & -B_k^T \\ B_k & C_k \end{bmatrix}$ produced by Algorithm 4.1, then $\|X_k\|_2 \leq (1 + |2\beta - 1|)\|L_k\|_2$ and $\|L_k\|_2 \leq \left(1 + \frac{|2\beta-1|}{2\beta}\right)\|X_k\|_2$.

Proof. Simply notice that

$$X_k = L_k + \begin{pmatrix} (2\beta - 1)A_k & 0 \\ 0 & 0 \end{pmatrix} \Rightarrow \|X_k\|_2 \leq \|L_k\|_2 + |2\beta - 1|\|L_k\|_2,$$

and

$$L_k = X_k + \begin{pmatrix} (1 - 2\beta)A_k & 0 \\ 0 & 0 \end{pmatrix} \Rightarrow \|L_k\|_2 \leq \|X_k\|_2 + \frac{|2\beta - 1|}{2\beta}\|X_k\|_2.$$

This concludes the proof. ∎

Then, Lemma C.2 allows to bound the higher order terms from (4.14) in terms of the norm of $\|A_k - A_{k+1}\|_2$.

Lemma C.2. In (4.14), we can bound the higher order terms by $\|\mathrm{H.O.T}_\Theta(4)\|_2 \leq |\tau|(|\tau|\delta)^2 \log\left(\frac{1}{1-2|\tau|\delta}\right)\|A_k - A_{k+1}\|_2$.

Proof. We know from the commutator version of the Goldberg series [24] (or BCH formula) that

$$\|\mathrm{H.O.T}_\Theta(4)\|_2 \leq \sum_{l=4}^{\infty}\sum_{w_l}\frac{|g_{w_l}|}{l}\|[w_l(\tau A_k, \tau A_{k+1})]\|_2,$$

where $w_l(\tau A_k, \tau A_{k+1})$ is a word of length l in the alphabet $\{\tau A_k, \tau A_{k+1}\}$ and the notation $[w_l(\tau A_k, \tau A_{k+1})]$, $[w_l]$ for short, is the extended commutator defined on this word [24]. g_{w_l} is the Goldberg coefficient associated to w_l [16]. Since $\|[A, B]\|_2 \leq 2\|A\|_2\|B\|_2$ and $\|[A, B]\|_2 \leq 2\|A\|_2\|B - A\|_2$, it follows by recurrence that $\|[w_l]\|_2 \leq 2^{l-1}\delta^{l-1}\tau^l\|A_k - A_{k+1}\|_2$. Moreover, for words of length l, we have $\sum_{w_l}\frac{|g_{w_l}|}{l} \leq \frac{2}{l}$ [34]. [35, Lem. A.1] even decreased this bound to $\sum_{w_l}\frac{|g_{w_l}|}{l} \leq \frac{1}{l}$. It follows that

$$\begin{aligned}\|\mathrm{H.O.T}_\Theta(4)\|_2 &\leq \sum_{l=4}^{\infty}\sum_{w_l}\frac{|g_{w_l}|}{l}2^{l-1}\delta^{l-1}\tau^l\|A_k - A_{k+1}\|_2 \\ &\leq |\tau|(|\tau|\delta)^2\|A_k - A_{k+1}\|_2\sum_{l=1}^{\infty}(2|\tau|\delta)^l\left(\sum_{w_{l+3}}\frac{|g_{w_{l+3}}|}{l+3}\right)\end{aligned}$$

$$\|\text{H.O.T}_\Theta(4)\|_2 \leq |\tau|(|\tau|\delta)^2\|A_k - A_{k+1}\|_2 \sum_{l=1}^{\infty} \frac{(2|\tau|\delta)^l}{l}$$
$$\leq |\tau|(|\tau|\delta)^2 \log\left(\frac{1}{1-2|\tau|\delta}\right)\|A_k - A_{k+1}\|_2,$$

where $2|\tau|\delta < 1$ stands by Condition 4.4. ∎

Lemma C.3 shows that it is indeed valid to go from (4.15) to (4.16) in the proof of Theorem 4.3.

Lemma C.3. Equation (4.16) follows from (4.15).

Proof. The top-left $p \times p$ block of the BCH series expansion of A_{k+1} based on the fundamental equation (4.3) yields

$$\begin{aligned}(\text{C.1}) \quad 2\beta(A_{k+1} - A_k) &= \Theta_k + \beta[A_k, \Theta_k] \\ &+ \frac{1}{12}\Big(4\beta^2[A_k,[A_k,\Theta_k]] - B_k^T B_k \Theta_k - \Theta_k B_k^T B_k + 2B_k^T \Gamma_k B_k \\ &- 2\beta[\Theta_k,[A_k,\Theta_k]]\Big) - \frac{1}{24}\Big(2[\Theta_k, B_k^T \Gamma_k B_k]\Big) + \mathcal{O}(\|\Theta_k\|_2^2) \\ &+ \text{H.O.T}_A(5).\end{aligned}$$

Equation (C.1) features many terms that we tackle one by one. All the terms can be bounded by leveraging (4.15) and $\|\Gamma_k\|_2 \leq \frac{6}{6-\delta^2}\|C_k\|_2 \leq \frac{6\delta}{6-\delta^2}$ [36]:

- $\|[A_k, \Theta_k]\|_2 \leq 2\|A_k\|_2\|\Theta_k\|_2 \leq 2\delta\eta\|A_k - A_{k+1}\|_2$.
- $\|[A_k,[A_k,\Theta_k]]\|_2 \leq 2\|A_k\|_2\|[A_k,\Theta_k]\|_2 \leq 4\delta^2\eta\|A_k - A_{k+1}\|_2$.
- $\|B_k^T B_k \Theta_k\|_2 \leq \delta^2\eta\|A_k - A_{k+1}\|_2$.
- $\|B_k^T \Gamma_k B_k\|_2 \leq \frac{6\delta^2}{6-\delta^2}\|C_k\|_2$.
- $\|[\Theta_k,[A_k,\Theta_k]]\| \in \mathcal{O}(\|A_k - A_{k+1}\|_2^2)$.
- $\|[\Theta_k, B_k^T \Gamma_k B_k]\|_2 \leq 2\delta^2\|\Gamma_k\|_2\|\Theta_k\|_2 \leq \frac{12\delta^3}{6-\delta^2}\eta\|A_k - A_{k+1}\|_2$.
- $\mathcal{O}(\|\Theta_k\|_2^2) \in \mathcal{O}(\|A_k - A_{k+1}\|_2^2)$

Inserting all these terms in (C.1) yields (4.16). ∎

Appendix D. An alternative algorithm to solve Problem 4.6. Algorithm D.1 proposes a shooting method on $\text{SO}(2p)$ to solve Problem 4.6, inspired from [9, Algorithm 1]. In Problem 4.6, notice that the initial shooting direction is $M := \dot{\gamma}(0) = \begin{bmatrix} D & -B^T \\ B & C \end{bmatrix}$. Starting from an initial guess for $M_0 = \dot{\gamma}_0(0)$, M_k is updated using an approximate parallel transport of the error vector $\Delta_k := V - \gamma_k(1)$ along the curve γ_k. We follow the method of [9, Algorithm 1] to approximate this parallel transport of Δ_k to the tangent space of $\gamma_k(0) = I_{2p}$, written $T_{I_{2p}}\text{SO}(2p)$: Δ_k is sequentially projected on $T_{\gamma_k(t)}\text{SO}(2p)$ for $t \in [t_m, t_{m-1}, ..., t_1]$ with t_m=1 and $t_1 = 0$. The pseudo-code of the method is provided in Algorithm D.1.

Appendix E. Logistic model fitting. We obtain a probabilistic radius of convergence of Algorithm 4.1 from numerical experiments. First, define a function $\mathcal{X}_\beta : \text{St}(n,p) \times \text{St}(n,p) \mapsto \{0,1\}$ where

$$\mathcal{X}_\beta(U, \widetilde{U}) = \begin{cases} 1 \text{ if Algorithm 4.1 converged with } (U, \widetilde{U}, \beta) \text{ as input.} \\ 0 \text{ otherwise.} \end{cases}$$

Algorithm D.1 The subproblem's shooting algorithm

INPUT: Given $V \in \mathrm{SO}(2p)$, $\beta > 0$, $\varepsilon > 0$ and $[t_1, t_2, ..., t_m]$ with $t_1 = 0$ and $t_m = 1$.
Initialize $M_0 := \begin{bmatrix} D_0 & -B_0^T \\ B_0 & C_0 \end{bmatrix}$ and $k = 0$.
while $\nu > \epsilon$ **do**
  for $j = m, m-1, ..., 1$ **do**
    $V^s \leftarrow \exp\left(t_j \begin{bmatrix} 2\beta D_k & -B_k^T \\ B_k & C_k \end{bmatrix}\right) \exp\left(t_j \begin{bmatrix} (1-2\beta)D_k & 0 \\ 0 & 0 \end{bmatrix}\right)$
    if $j == m$ **then**
      $W \leftarrow V - V^s$
      $\nu \leftarrow \|\Delta\|_{\mathrm{F}}$
    end if
    $M^s \leftarrow \mathrm{skew}((V^s)^T W)$ #Project W on $T_{V^s}\mathrm{SO}(2p)$.
    $W \leftarrow V^s M^s \cdot \frac{\nu}{\|M^s\|_{\mathrm{F}}}$
  end for
  $M_k \leftarrow M_k + M^s$, $k = k + 1$.
end while
return D_k

Given $\beta > 0$ and N samples $\{U_i, \widetilde{U}_i\}_{i\in\{1...,N\}}$ drawn from a continuous distribution on $\mathrm{St}(n,p)$, we build a data set of pairs

$$\{x_i, y_i\}_{i\in\{1...,N\}} := \{\|U_i - \widetilde{U}_i\|_{\mathrm{F}},\ \mathcal{X}_\beta(U_i, \widetilde{U}_i)\}_{i\in\{1...,N\}}.$$

We expect Algorithm 4.1 to converge when $\|U_i - \widetilde{U}_i\|_{\mathrm{F}}$ is small and failures to appear when $\|U_i - \widetilde{U}_i\|_{\mathrm{F}}$ gets larger. This framework is natural to fit a logistic regression model $m_\theta : \mathbb{R} \mapsto (0,1)$, where $\theta := [\theta_0,\ \theta_1] \in \mathbb{R}^2$ is the fitting parameter. The logistic model predicts the probability of convergence of Algorithm 4.1. It is given by

$$m_\theta(x) = \frac{1}{1 + \exp(\theta_0 + \theta_1 x)}.$$

The estimator θ is chosen to maximize the likelihood of the data set, i.e.,

$$\theta := \arg\max_{\theta\in\mathbb{R}^2} \prod_{i=1}^{N} m_\theta(x_i)^{y_i}(1 - m_\theta(x_i))^{(1-y_i)}.$$

For reference, see, e.g., [21]. It is well-known that it is easier and equivalent to obtain the log-likelihood estimator

$$\theta := \arg\max_{\theta\in\mathbb{R}^2} f(\theta) := \arg\max_{\theta\in\mathbb{R}^2} \sum_{i=1}^{N} \log(m_\theta(x_i))y_i + \log(1 - m_\theta(x_i))(1 - y_i). \tag{E.1}$$

Equation (E.1) is a Lipschitz-smooth convex optimization problem and we solve it using the accelerated gradient method [23] with stopping criterion $\|\nabla f(\theta)\|_{\mathrm{F}} < 10^{-8}$. We considered $N = 1000$ in our experiments. The goodness of fit is confirmed by the high coefficients of determination R^2 provided in the caption of Fig. 7.

REFERENCES

[1] Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ (2008), http://press.princeton.edu/titles/8586.html

[2] Absil, P.-A., Mataigne, S.: The ultimate upper bound on the injectivity radius of the Stiefel manifold (2024), https://arxiv.org/abs/2403.02079

[3] Bendokat, T., Zimmermann, R.: Efficient quasi-geodesics on the Stiefel manifold pp. 763–771 (2021), https://doi.org/10.1007/978-3-030-80209-7

[4] Bergmann, R.: Manopt.jl: Optimization on manifolds in Julia. Journal of Open Source Software **7**(70), 3866 (2022), https://doi.org/10.21105/joss.03866

[5] Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. SIAM review **59**(1), 65–98 (2017), https://doi.org/10.1137/141000671

[6] Bhatia, R.: Matrix Analysis, vol. 169. Springer (1997)

[7] Boumal, N., Mishra, B., Absil, P.-A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. Journal of Machine Learning Research **15**(42), 1455–1459 (2014), http://jmlr.org/papers/v15/boumal14a.html

[8] Brigant, A.L., Puechmorel, S.: Quantization and clustering on Riemannian manifolds with an application to air traffic analysis. Journal of Multivariate Analysis **173**, 685–703 (2019), https://doi.org/10.1016/j.jmva.2019.05.008

[9] Bryner, D.: Endpoint Geodesics on the Stiefel Manifold Embedded in Euclidean space. SIAM Journal on Matrix Analysis and Applications **38**(4), 1139–1159 (2017), https://doi.org/10.1137/16M1103099

[10] do Carmo, M.P.: Riemannian Geometry. Mathematics: Theory & Applications, Birkhäuser Boston (1992), https://books.google.de/books?id=ct91XCWkWEUC

[11] Chakraborty, R., Vemuri, B.C.: Statistics on the Stiefel manifold: Theory and applications. The Annals of Statistics **47**(1), 415 – 438 (2019), https://doi.org/10.1214/18-AOS1692

[12] Cheeger, J.: Some examples of manifolds of nonnegative curvature. Journal of Differential Geometry **8**(4), 623–628 (Dec 1973), https://doi.org/10.4310/jdg/1214431964

[13] Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications **20**(2), 303–353 (1998), https://doi.org/10.1137/S0895479895290954

[14] Fréchet, M.R.: Les éléments aléatoires de nature quelconque dans un espace distancié. Annales de l'institut Henri Poincaré **10**(4), 215–310 (1948)

[15] Gao, B., Vary, S., Ablin, P., Absil, P.-A.: Optimization flows landing on the Stiefel manifold. IFAC-PapersOnLine **55**(30), 25–30 (2022), https://www.sciencedirect.com/science/article/pii/S2405896322026519, 25th IFAC Symposium on Mathematical Theory of Networks and Systems MTNS 2022

[16] Goldberg, K.: The formal power series for $\log e^x e^y$. Duke Math. J. **23**(1), 13–21 (1956), http://dx.doi.org/10.1215/S0012-7094-56-02302-X

[17] Huang, L., Liu, X., Lang, B., Yu, A., Wang, Y., Li, B.: Orthogonal weight normalization: Solution to Optimization over multiple dependent Stiefel Manifolds in Deep Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence **32**(1) (Apr 2018), https://ojs.aaai.org/index.php/AAAI/article/view/11768

[18] Hüper, K., Markina, I., Silva, L.F.: A Lagrangian approach to extremal curves on Stiefel manifolds. Journal of Geometric Mechanics **13**(1), 55–72 (2021), https://doi.org/10.3934/jgm.2020031

[19] Jung, S., Dryden, I., Marron, J.S.: Analysis of principal nested spheres. Biometrika **99**(3), 551–568 (2012), https://doi.org/10.1093/biomet/ass022

[20] Kent, J., Hamelryck, T.: Using the Fisher-Bingham distribution in stochastic models for protein structure. Quantitative Biology, Shape Analysis, and Wavelets **24**(1), 57–60 (2005)

[21] Maalouf, M.: Logistic regression in data analysis: An overview. International Journal of Data Analysis Techniques and Strategies **3**, 281–299 (07 2011), https://doi.org/10.1504/IJDATS.2011.041335

[22] Miolane, N., Guigui, N., Brigant, A.L., Mathe, J., Hou, B., Thanwerdas, Y., Heyder, S., Peltre, O., Koep, N., Zaatiti, H., Hajri, H., Cabanes, Y., Gerald, T., Chauchat, P., Shewmake, C., Brooks, D., Kainz, B., Donnat, C., Holmes, S., Pennec, X.: Geomstats: A Python Package for Riemannian Geometry in Machine Learning. Journal of Machine Learning Research **21**(223), 1–9 (2020), http://jmlr.org/papers/v21/19-027.html

[23] Nesterov, Y.: A method for solving the convex programming problem with convergence rate O(1/(k*k)). Proceedings of the USSR Academy of Sciences **269**, 543–547 (1983)

[24] Newman, M., Thompson, R.C.: Numerical values of Goldberg's coefficients in the series for $\log(e^x e^y)$. Mathematics of Computation **48**(177), 265–s132 (1987), http://www.jstor.org/

stable/2007889
[25] Nguyen, D.: Curvatures of Stiefel manifolds with deformation metrics. Journal of Lie Theory **32**(2), 563–600 (2022), https://arxiv.org/abs/2105.01834
[26] Nguyen, D.: Closed-form geodesics and optimization for Riemannian logarithms of Stiefel and flag manifolds. Journal of Optimization Theory and Applications **194**(1), 142–166 (2022), https://doi.org/10.1007/s10957-022-02012-3
[27] Noakes, L.: A global algorithm for geodesics. Journal of the Australian Mathematical Society. Series A. Pure Mathematics and Statistics **65**(1), 37–50 (1998), https://doi.org/10.1017/S1446788700039380
[28] Pennec, X., Fillard, P., Ayache, N.: A Riemannian Framework for Tensor Computing. International Journal of Computer Vision **66**(1), 41–66 (1 2006), https://doi.org/10.1007/s11263-005-3222-z
[29] Rentmeesters, Q.: Algorithms for data fitting on some common homogeneous spaces. Ph.D. thesis, Catholic University of Louvain (UCLouvain) (2013), https://dial.uclouvain.be/pr/boreal/fr/object/boreal:132587
[30] Sakai, T.: Riemannian Geometry. Fields Institute Communications, American Mathematical Society (1996), https://books.google.be/books?id=ODDyngEACAAJ
[31] Stoye, J., Zimmermann, R.: On the injectivity radius of the Stiefel manifold: Numerical investigations and an explicit construction of a cut point at short distance (2024), https://arxiv.org/abs/2403.03782
[32] Sutti, M.: Shooting methods for computing geodesics on the Stiefel manifold (2023), https://arxiv.org/abs/2309.03585
[33] Sutti, M., Vandereycken, B.: The leapfrog algorithm as nonlinear Gauss-Seidel (2023), https://arxiv.org/abs/2010.14137
[34] Thompson, R.C.: Convergence proof for Goldberg's exponential series. Linear Algebra and its Applications **121**, 3–7 (1989). https://doi.org/https://doi.org/10.1016/0024-3795(89)90688-5, https://www.sciencedirect.com/science/article/pii/0024379589906885
[35] Zimmermann, R.: A Matrix-Algebraic Algorithm for the Riemannian Logarithm on the Stiefel Manifold under the Canonical Metric. SIAM Journal on Matrix Analysis and Applications **38**(2), 322–342 (2017), https://doi.org/10.1137/16M1074485
[36] Zimmermann, R., Hüper, K.: Computing the Riemannian Logarithm on the Stiefel Manifold: Metrics, Methods, and Performance. SIAM Journal on Matrix Analysis and Applications **43**(2), 953–980 (2022), https://doi.org/10.1137/21M1425426