Introduction to Riemannian Geometry and Geometric Statistics: from basic theory to implementation with Geomstats

Nicolas Guigui*, Nina Miolane†, Xavier Pennec*

October 2022

Contents

*Université Côte d'Azur, Inria, Epione project team, France
†University of California Santa Barbara, USA

List of Figures

List of Examples

1 Introduction

Since the formal axiomization of Euclid in his famous *Elements* (dated around 300 BC), geometry was considered as the properties of figures in the plane or in space. The abstract notion of space as a mathematical object emerged in 1827 with C. F. Gauss' *Theorema Egrerium* proving that curvature is an intrinsic quantity of a surface, i.e. that can be computed without reference to a "larger" embedding space. This notion was made precise by the cornerstone work of Riemann 1868[1] built around the intuitive idea that a mathematical space results from varying a number of independent quantities, later identified as coordinates and formalized in the definition of a manifold by Whitney 1936. Riemannian Geometry (RG) is the study of such differentiable manifolds equipped with an inner product at each point that smoothly varies between points. This allows us to generalize the notions of angles, length and volumes, which can be integrated to global quantities highly coupled with the topology of the space.

Fruitful developments of these ideas allowed unifying previous examples of non-Euclidean geometries, that violate Euclid's parallel postulate (given a point and a straight-line, one and only one parallel straight-line can be drawn through the point). These ideas echoed with the developments of Lagrangian and Hamiltonian mechanics, and were instrumental in formalizing the modern theories of Physics, and especially Einstein's general relativity. They also made profound impact on many areas of mathematics such as group theory, representation theory, analysis, and algebraic and differential topology.

At the intersection of Physics and geometry, groups represent symmetries and transformations between states, and from the modern point of view of Klein's Erlangen programm, the study of geometry boils down to studying the action of groups on a space, and their invariants. The work of Elie Cartan enabled significant progress in this direction.

Riemannian geometry has thus become a vast subject that is not usually taught before graduate education in mathematics or physics, and that requires familiarity with many concepts from differential geometry. Hence, although some books on the topic cover most of the pre-requisites and fundamental results of Riemannian geometry, the entry cost for applied mathematicians, computer scientists and engineers is high.

Nowadays, as data is a predominant resource in applications, Riemannian geometry is a natural framework to model and unify complex nonlinear sources of data. However, the development of computational tools from the basic theory of Riemannian geometry is laborious due to often high dimensional and non-exhaustive coordinate systems, nonlinear and intractable differential equations, etc. This monograph aims at providing the computational tools to perform statistics and machine learning on Riemannian manifolds to the wider data science community. The work presented here forms one of the main contributions to the open-source project geomstats, that consists in a Python package providing efficient implementations of the concepts of Riemannian geometry and geometric statistics, both for mathematicians and for applied scientists for whom most of the difficulties are hidden under high-level functions.

Other Python packages do exist and mainly fall under one of two following categories. On the one hand, there are the packages that focus on a single application, for instance on optimization: `Pymanopt` (Townsend, Koep, et al. 2016), `Geoopt` (Becigneul and Ganea 2019; Kochurov 2019), `TensorFlow RiemOpt` Smirnov 2021, and `McTorch` (Meghwanshi, Jawanpuria, et al. 2018) or on deep learning such as `PyG` (Fey and Lenssen 2019), where, in this case the geometry is often restricted to graph and mesh spaces. On the other hand, there are packages dedicated to a single manifold: `PyRiemann` on SPD matrices (Barachant 2015), `PyQuaternions` on 3D rotations (Wynn 2014), and `PyGeometry` on spheres, tori, 3D rotations and translations (Censi 2012). Some other packages, like `TheanoGeometry` (Kühnel and Sommer 2017) are not actively maintained anymore. There is therefore a need for a unified open-source implementation of differential geometry and associated learning algorithms for manifold-valued data.

[1] Riemann 1873, English Translation by W. K. Clifford.

The goal of this monograph is two-fold. First, we aim at giving a self-contained exposition of the basic concepts of Riemannian geometry, providing illustrations and examples at each step and adopting a computational point of view. We cover the basics of differentiable manifolds (Section 2), Riemannian manifolds (Section 3) and Lie groups (Section 4). Then we delve into more complexe structures defined by invariance properties, in particular quotient metrics, and metrics on homogeneous and symmetric spaces (Section 5). Most proofs are omitted for brevity, but references to the proof of each statement are given. The interested reader may refer to the textbooks Lafontaine, Gallot, et al. 2004; Gallier and Quaintance 2020; Boumal 2022; Lee 2003 for more details. Some mathematical definitions from the prerequisites can be found in the lexicon in Appendix A. The second goal is to demonstrate how these concepts are implemented in geomstats, explaining the choices that were made and the conventions chosen. The general concepts are exposed in Subsection 2.2, and detailed along the text and examples. The culmination of this implementation is to be able to perform statistics and machine learning on manifolds, with as few lines of codes as in the wide-spread machine learning tool `scikit-learn`. We exemplify this in Section 6 with a brief introduction to geometric statistics.

2 Differentiable manifolds

2.1 Differentiable manifolds and tangent spaces

The differentiable manifold will be the structure underlying this entire monograph, yet its definition remains difficult for newcomers in the field. We start by that of an embedded manifold and generalize to the abstract case in a second part. The intuition behind the notion of smooth manifold is that around every point, it resembles the d-dimensional vector space $\mathbb{R}^d$ for some integer d, and the properties of $\mathbb{R}^d$ allow to define the notions of smooth functions, tangent vectors, etc. on the manifold.

2.1.1 Embedded manifolds

Let N be a strictly positive integer. The fundamental examples of an embedded manifold are that of an open set* of $\mathbb{R}^N$, and a vector subspace $\mathbb{R}^d \times \{0\}_{N-d} \subset \mathbb{R}^N$ for $d \leq N$, written $\mathbb{R}^d \times 0$ for short. These are 'deformed' via local diffeomorphisms* to obtain an embedded manifold, which can be thought of as a smooth surface in the ambient space. This was in fact one of the first motivations of the mathematical developments underlying the notion of manifold.

Definition 2.1. Let d, N be integers with $1 \leq d \leq N$. Then a d-dimensional smooth *embedded manifold* in $\mathbb{R}^N$ is a non-empty subset M of $\mathbb{R}^N$ such that for every point $p \in M$, there are two open subsets $U, V \subseteq \mathbb{R}^N$ with $p \in U$ and $0 \in V$, and a smooth diffeomorphism* $\varphi : U \to V$ such that $\varphi(U \cap M) = V \cap (\mathbb{R}^d \times \{0\}_{N-d})$.

In this definition, φ may be called a local *chart*. Thankfully there are equivalent definitions that give greater insights into what makes M a differentiable manifoldWe first need to define the notions of immersions and submersions.

Definition 2.2. Let $n \leq p$ be two strictly positive integers, and $U \in \mathbb{R}^p, V \in \mathbb{R}^n$ two open sets.

- A differentiable* map $f : U \to V$ is called a *submersion* at $x \in U$ if df_x is surjective*. We say that f is a submersion if it is a submersion at every $x \in U$.
- A differentiable map $f : V \to U$ is called an *immersion* at $x \in V$ if df_x is injective*. We say that f is an immersion if it is an immersion at every $x \in V$.

*Defined in Appendix A.

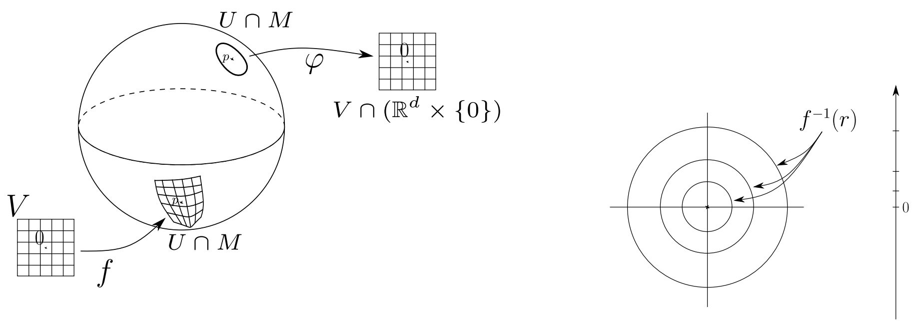


Figure 1: Representation of a manifold.
Left: Representation of a manifold, definition with local φ or by a local immersion f. Right: Representation of a manifold defined by a submersion, in this case the distance to the origin.

The fundamental example of an immersion is the injection defined by $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, 0, \ldots, 0) \in \mathbb{R}^p$, while the projection $(x_1, \ldots, x_p) \in \mathbb{R}^p \mapsto (x_1, \ldots, x_n) \in \mathbb{R}^n$ is that of a submersion. In fact, one can show that up to a local change of variables (i.e. composition with a diffeomorphism), these maps are respectively the only immersions and submersions. This results from the local inversion theorem, see Balzin n.d., Theorem 4.7 for a proof. We now have the following characterisation theorem (Gallier and Quaintance 2020, Theorem 3.6):

Theorem 2.1. *A nonempty subset $M \subseteq \mathbb{R}^N$ is a d-dimensional manifold if and only if any of the following conditions hold:*

(1) (Local parametrization) For every $p \in M$, there are two open subsets $V \subseteq \mathbb{R}^d$ and $U \subseteq \mathbb{R}^N$ with $p \in U$ and $0 \in V$, and a smooth function $f : V \to \mathbb{R}^N$ such that $f(0) = p$, f is a homeomorphism between V and $U \cap M$, and f is an immersion at 0.*

(2) (Local implicit function) For every $p \in M$, there exist an open set $U \in \mathbb{R}^N$ and a smooth map $f : U \to \mathbb{R}^{N-d}$ that is a submersion at p, such that $U \cap M = f^{-1}(\{0\})$[2].

(3) (Local graph) For every $x \in M$, there exist an open neighborhood $U \subseteq \mathbb{R}^N$ of x, a neighborhood $V \subseteq \mathbb{R}^d$ of 0 and a smooth map $f : V \to \mathbb{R}^{N-d}$ such that $U \cap M = \mathrm{graph}(f)$[3].

The characterization (2) encodes the notion of constraint: a manifold can be understood as the set of points that verify a constraint defined by an implicit equation, given by the function f. This is one of the reasons manifolds are ubiquitous in applications, we will give many examples of this case. The other characterizations can also be understood as follows. The first (1) implies that at every point of the manifold, a coordinate system defined on $\mathbb{R}^d$ exists to parametrize the manifold around that point. Finally (3) is the most common to think of surfaces in $\mathbb{R}^3$ as sets of points $(x, y, f(x, y))$.

Example 2.1: Hypersphere

The most simple manifold we will study is the hypersphere, or d-dimensional sphere. It is the set of unit-norm vectors of $\mathbb{R}^{d+1}$:

$$S^d = \{x \in \mathbb{R}^{d+1} \mid \|x\|_2^2 = 1\}.$$

Let $f : x \in \mathbb{R}^{d+1} \mapsto \|x\|^2 - 1 \in \mathbb{R}$. For $x \neq 0$, $df_x y = 2x^\top y$ is surjective, so

[2] Recall that the *pre-image* of a set A by $f : E \to F$ is defined by $f^{-1}(A) = \{x \in E, f(x) \in A\}$.
[3] the graph of f is the set $\mathrm{graph}(f) = \{(x, f(x)) | x \in V\}$.

Assertion (2) of Theorem 2.1 applies (with $U = \mathbb{R}^{d+1}$) and S^d is a d-dimensional embedded manifold in $\mathbb{R}^{d+1}$. In dimension $d = 1$, this corresponds to the circle, and for $d = 2$ this is the usual sphere. Both cases are common to represent angles and directions in space, and as such appear in the field of directional statistics (Mardia and Jupp 2009).

Example 2.2: Hyperbolic space

The fundamental counterpart to the hypersphere is the two-sheeted hyperboloid, defined by

$$H^d = \{x \in \mathbb{R}^{d+1} \mid -x_0^2 + \sum_{i=1}^{d} x_i^2 = -1\}.$$

It is one of the models of hyperbolic geometry, which is increasingly used to model hierarchical data, e.g. (Nickel and Kiela 2017).

Example 2.3: Special Orthogonal group

Matrix groups play an essential role in the theory of RG, and especially the special orthogonal group $SO(n)$, i.e. the set of unit determinant orthogonal matrices:

$$SO(n) = \{R \in M_n(\mathbb{R}) \mid R^\top R = I_n, \quad \det(R) = 1\}.$$

Consider the map $f : \left\{ \begin{array}{ccc} GL^+(n) & \longrightarrow & S(n) \\ A & \longmapsto & A^\top A - I_n \end{array} \right.$ where $GL^+(n) \in M_n(\mathbb{R}) \simeq \mathbb{R}^{n^2}$ is the open set of invertible squared matrices with positive determinant, $S(n)$ is the set of symmetric matrices of size n, a vector subspace of $M_n(\mathbb{R})$ of dimension $\frac{n(n+1)}{2}$. It is straightforward to show that the differential of f at some R is

$$df_R(H) = R^\top H + H^\top R,$$

and we can see that it is surjective for all $R \in SO(n)$, as for any $S \in S(n)$, $df_R\left(\frac{RS}{2}\right) = S$. As $SO(n) = f^{-1}(0)$, we conclude that $SO(n)$ is indeed an embedded manifold of dimension $\frac{n(n-1)}{2}$.

One can represent the motion of a rigid-body in the referential of its barycenter as a curve with values in $SO(3)$, hence this group is widely used in e.g. robotics (Barczyk, Bonnabel, et al. 2015).

Example 2.4: Stiefel manifold

A generalization of both hypersphere and special orthogonal group is the Stiefel manifold, defined as the set of orthonormal k-frames of $\mathbb{R}^n$. If we represent each vector u_i of a k-frame $(u_1, \dots, u_k)$ as the i^{th} column of a matrix U (in the canonical basis of $\mathbb{R}^n$), then the Stiefel manifold can be seen as a subset in $M_{n,k}(\mathbb{R})$:

$$St(k, n) = \{U \in M_{n,k}(\mathbb{R}) \mid U^\top U = I_k\}.$$

As in the previous example, we can consider the map $f : U \mapsto U^\top U - I$ on an open subset of $M_{n,k}(\mathbb{R})$ and show that it is a submersion such that $S(k, n) = f^{-1}(0)$ to conclude that $S(k, n)$ is an embedded manifold of dimension $nk - \frac{k(k+1)}{2}$.

The Stiefel manifold naturally arises as the optimization domain in many problems related to matrix decompositions, in linear algebra, statistics, machine learning, computer vision, etc. see Absil, Mahony, et al. 2010, and references therein.

2.1.2 Manifolds

For generality, we now define a manifold in a more abstract way, i.e. as a topological space *that is not a priori embedded in some $\mathbb{R}^N$. The idea is still that a manifold is a space that can be covered by open sets that each look like (i.e. are diffeomorphic to an open set of) the usual space $\mathbb{R}^d$. Of course one can verify that embedded manifolds are indeed manifolds with this more general definition, and in fact, Whitney 1936 proved that any manifold can be smoothly embedded in a larger space, showing that the two concepts are equivalent. We first motivate the need for a more general definition of manifold by the example of the Kendall size-and-shape space.

Example 2.5: Kendall size-and-shape space

The underlying idea is that a shape is what is left after removing the effects of translation and rotation. We first define the set of k landmarks of $\mathbb{R}^m$ as the space of $m \times k$ matrices $M_{m,k}(\mathbb{R})$. For $x \in M_{m,k}(\mathbb{R})$, let x_i denote the columns of x, i.e. points of $\mathbb{R}^m$ and let $\bar{x}$ be their barycenter. We remove the effects of translation by considering the matrix with columns $x_i - \bar{x}$ instead of x. Let $M^*_{m,k}(\mathbb{R})$ be the set of such centered matrices.

Now, in order to remove the effects of rotations, we would like to *identify* the landmark configurations that only differ by a rotation of all the landmarks. This defines an equivalence relation $\sim$:

$$\forall x, y \in M^*_{m,k}(\mathbb{R}), \quad x \sim y \iff \exists R \in SO(m), y = Rx.$$

A shape thus corresponds to an equivalence class $[x]$ of landmark configurations, and we can define the size-and-shape space as the quotient of the landmark space by the equivalence relation $\sim$ (or equivalently by $SO(n)$):

$$S\Sigma^k_m = \{[x] \mid x \in M^*_{m,k}(\mathbb{R})\}.$$

A quotient space of a manifold by another manifold may not even be a Hausdorff space, but we will give sufficient conditions in Section 4.4 to ensure that quotients resulting from a group action are indeed smooth manifolds.

In this case, $S\Sigma^k_m$ inherits a differentiable structure from the landmark space, that turns it into a smooth manifold, although we cannot see it explicitly as a subset of $\mathbb{R}^N$ for some N. Implementing tools to work with data on such spaces is a challenging task that we tackle in geomstats. This is the subject of section 5. Kendall size-and-shape spaces are a ubiquitous framework for the statistical analysis of data arising from medical imaging, computer vision, biology, chemistry and many more domains (I. L. Dryden and Mardia 2016).

Definition 2.3 (Atlas)**.** Let M be a topological space* and $k \geq 1$ an integer. A C^k*-atlas* $\mathcal{A}$ with values in $\mathbb{R}^d$ is a collection of pairs (U, φ) called *charts* where $\varphi : U \to V$ is a homeomorphism between the open sets $U \subset M$ and $V \subset R^d$, such that $M \subset \bigcup_{U \in \mathcal{A}} U$ and for any (U, φ) and (U', φ') in $\mathcal{A}$, the *transition map*

$$\varphi' \circ \varphi^{-1} : \varphi(U \cap U') \to \varphi'(U \cap U')$$

*Defined in Appendix A.

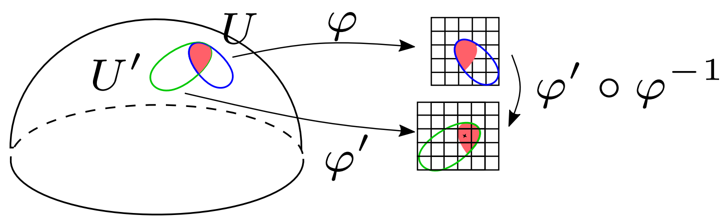


Figure 2: Transition maps

is a C^k-diffeomorphism*.

Two atlases are C^k-*compatible* if their union is still a C^k-atlas. Compatibility defines an equivalence relation, and we will think of the equivalence class of an atlas whenever referring to one. There is a unique *maximal* atlas (for the inclusion) that contains a given atlas.

Note that the transition maps are defined between open sets of $\mathbb{R}^d$ (see Figure 2), the usual notions of differentiability are thus available, and will allow to define such notions on manifolds. We will always consider the case $k = \infty$ and say that C^∞ maps and atlases are *smooth*.

Definition 2.4 (Differentiable manifold)**.** We call *differentiable manifold* of class C^k and dimension d any topological space M that is Hausdorff* and second-countable* together with a maximal C^k-atlas $\mathcal{A}$ with values in $\mathbb{R}^d$.

We sometimes refer to the atlas of M as its differentiable structure, and to this definition of manifold as the *intrinsic* definition as the charts are defined on M rather than an extrinsic embedding space. A chart (U, φ) defines a set of *local coordinates* on U written $(x^1, \dots, x^d)$ for short, and defined by $x^i = pr_i(\varphi(x))$, where pr_i is the projection on the i^{th} coordinate of $\mathbb{R}^d$. let us now use the intrinsic definition of a manifold to exemplify further how manifolds can be obtained from others.

Example 2.6: Product manifold

Let M, N be two manifolds with $(U_i, \varphi_i)_{i\in I}, (V_j, \psi_j)_{j\in J}$ their respective atlas. Define for $(i, j) \in (I \times J)$

$$\phi_{ij} : \left\{ \begin{array}{ccc} U_i \times V_j & \longrightarrow & \varphi_i(U_i) \times \psi_j(V_j) \\ (x, y) & \longmapsto & (\varphi_i(x), \psi_j(y)) \end{array} \right.$$

Then it is easy to check that $(U_i \times V_j, \varphi_{ij})_{(i,j)\in I\times J}$ is an atlas for the product space $M \times N$. This atlas does not depend on the choice of original atlases of M, N (in the right equivalence class), and allows to endow $M \times N$ with the structure of manifold. We call it the product manifold of M and N. Its dimension is $\dim(M \times N) = \dim M + \dim N$.

To give insights into the importance of the general notion of manifold, let us now consider two counter-examples.

Example 2.7: Cusp and Node

For more details on these two examples see Gallier and Quaintance 2020, chapter 7. First, we consider the classic example of a space that is not a manifold: the

*Defined in Appendix A.

nodal cubic, shown in Figure 3. It is the set of points

$$M_1 = \{(x, y) \in \mathbb{R}^2 \mid y^2 = x^2 - x^3\},$$

considered under the subset topology. The self-intersection at the origin does nt preserve the topology of $\mathbb{R}$, so no homeomorphism can exist between M and $\mathbb{R}$ around the origin. Thus, M is not a manifold.

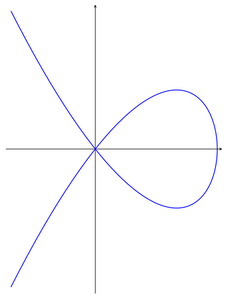


Figure 3: Nodal cubic M_1

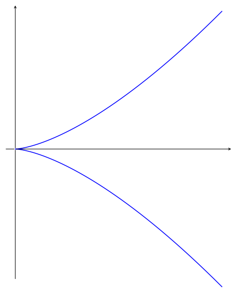


Figure 4: Cuspidal curve M_2

Secondly, we consider the cuspidal curve displayed on Figure 4 and defined as the set

$$M_2 = \{(x, y) \in \mathbb{R}^2 \mid y^2 = x^3\}.$$

We can define the maps $\varphi : (x, y) \in M_2 \mapsto y^{1/3} \in \mathbb{R}$ and $\psi : (x, y) \in M_2 \mapsto y \in \mathbb{R}$, that each define a smooth atlas on M_2 and endow it with a differentiable structure of manifold. However, the two atlases (constituted of a single chart) are not compatible, so they define different manifolds.

For the next notions that will be introduced, we will use the convenient setting of embedded manifolds, but all these notions can be generalized to the abstract case by using charts to recover functions defined between vector spaces.

2.1.3 Tangent spaces and differentiable maps

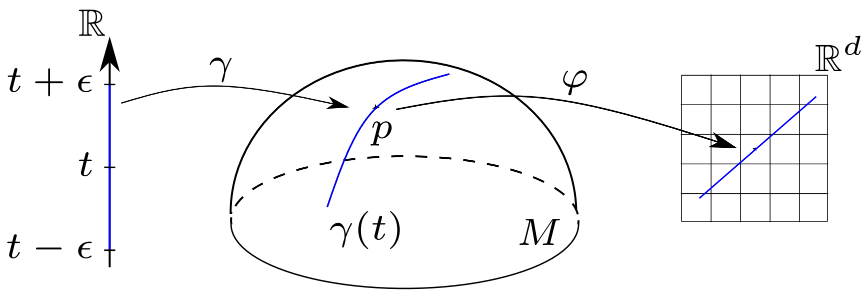


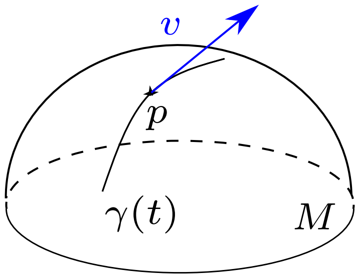
Figure 5: Definition of a smooth curve on a manifold.

We first define smooth curves on manifolds, using a local parametrization of M and the notion of smooth function from $\mathbb{R}$ to $\mathbb{R}^d$. See Figure 5 for a representation.

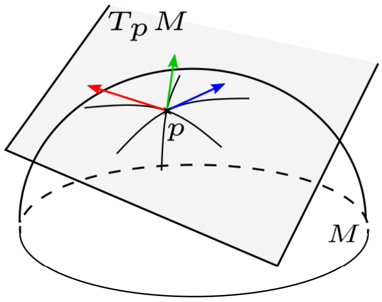
The following definition does not depend on the choice of local parametrization.

Definition 2.5 (Smooth curve)**.** Let M be a d-dimensional manifold in $\mathbb{R}^N$. A *smooth curve* γ in M is any function $\gamma : I \to M$ where I is an open interval in $\mathbb{R}$, such that for any $t \in I$, $p = \gamma(t)$, there is a local chart $\varphi : U \to V$ of M at p and $\epsilon > 0$ such that $\varphi \circ \gamma : (t - \epsilon, t + \epsilon) \to \mathbb{R}^d$ is smooth.

This definition is extended to smooth curves defined on a closed interval $I = [a, b]$ of $\mathbb{R}$ by requiring that γ be the restriction of some smooth curve defined on an open interval that contains $[a, b]$. As $\gamma : I \to M \subset \mathbb{R}^N$ is a curve in $\mathbb{R}^N$ and is differentiable, a tangent vector along γ at some time $t \in I$ is obviously defined. This generalizes to tangent spaces to the manifold.



(a) Definition of a tangent vector as the derivative of a curve.



(b) The collection of all tangent vectors forms a vector space.

Definition 2.6 (Tangent vector)**.** Let $d \leq N \in \mathbb{N}$, M be an embedded manifold in $\mathbb{R}^N$ of dimension d and $p \in M$. A vector $v \in \mathbb{R}^N$ is *tangent* to M at p if there exists an open interval I centered around 0, and a curve $\gamma : I \to M$ such that

$$\gamma(0) = p \text{ and } \dot{\gamma}(0) = v.$$

We write T_pM for the set of tangent vectors at p.

Recall that $\mathbb{R}^d \times \{0\}$ is a fundamental example of an embedded manifold, it is clear that the tangent space at any $p \in \mathbb{R}^d \times \{0\}$ is the whole $\mathbb{R}^d \times \{0\}$. From this case we deduce the characterizations of tangent spaces equivalent to that of manifolds obtained in Theorem 2.1 (Paulin 2007, Proposition 3.1).

Theorem 2.2. *Let M be a manifold in $\mathbb{R}^N$ of dimension d.*

(1) If $U, V \subset \mathbb{R}^N$ are two open neighborhoods respectively around p and 0 in $\mathbb{R}^N$ and $f : U \to V$ is a diffeomorphism such that $f(p) = 0$ and $f(U \cap M) = V \cap (\mathbb{R}^d \times \{0\})$ then

$$T_pM = df_p^{-1}(\mathbb{R}^d \times \{0\}).$$

(2) (Local parametrization) If $U \subseteq \mathbb{R}^N$ is an open neighborhood of $p \in M$, $V \subseteq \mathbb{R}^d$ is an open neighborhood around 0 and $f : V \to \mathbb{R}^N$ is a smooth function such that $f(0) = p$, f is a homeomorphism between V and $U \cap M$, and f is an immersion at 0, then[2]

$$T_pM = \operatorname{Im} df_0.$$

(3) (Local implicit function) If $U \subseteq \mathbb{R}^N$ is an open neighborhood around $p \in M$ and $f : U \to \mathbb{R}^{N-d}$ is a smooth map that is a submersion at p, such that $U \cap M = f^{-1}(\{0\})$, then[3]

$$T_pM = \ker df_p.$$

(4) (Local graph) If $U \subseteq \mathbb{R}^N$ is an open neighborhood of $p \in M$, $V \subseteq \mathbb{R}^d$ a neighborhood of 0 and $f : V \to \mathbb{R}^{N-d}$ a smooth map such that $U \cap M = \operatorname{graph}(f)$ and $p = (0, f(0))$, then

$$T_pM = \operatorname{Im}\{v \mapsto (v, df_0(v))\}.$$

[2]Recall that the *range* of $f : E \to F$ is defined by $\operatorname{Im} f = \{f(x), x \in E\} \subset F$.
[3]Recall that the *kernel* of $f : E \to F$ is defined by $\ker f = \{x \in E \mid f(x) = 0\} \subset E$.

From (1) we can see that T_pM is a linear subspace of $\mathbb{R}^N$ of dimension d. Tangent spaces thus provide local linearizations of the manifold, a property that will be useful as a first way to handle data on manifolds. The previous theorem allows computing the tangent spaces of the common manifolds seen in the previous section.

Example 2.8: Tangent space of the hypersphere

Recall that the hypersphere is the embedded manifold defined by $S^d = f^{-1}(0)$ where $f : x \mapsto \|x\|^2 - 1$. This corresponds to (3) of Theorem 2.2, therefore for any $x \in S^d$

$$T_xS^d = \{v \in \mathbb{R}^{d+1} \mid \langle x, v\rangle = 0\}.$$

Example 2.9: Tangent space of the hyperbolic space

Similarly, as the hyperbolic space is defined as $H^d = f^{-1}(0)$ where $f : x \mapsto -x_0^2 + \sum_{i=1}^d x_i^2 + 1$, we obtain for any $x \in H^d$

$$T_xH^d = \{v \in \mathbb{R}^{d+1} \mid -x_0v_0 + \sum_{i=1}^{d} x_iv_i = 0\}.$$

Example 2.10: Tangent space of the special orthogonal group

Recall $SO(n) = f^{-1}(0)$ with $f : A \mapsto A^\top A - I_n$, and for any $R \in SO(n), H \in M_n(\mathbb{R})$ we have $df_R(H) = R^\top H + H^\top R$. Therefore for any $R \in SO(n)$

$$T_RSO(n) = \{H \in M_n(\mathbb{R}) \mid R^\top H + H^\top R = 0\}.$$

Note the special case $R = I_n$, then $T_{I_n}SO(n) = \mathrm{Skew}(n)$, the set of skew-symmetric matrices of size n. The tangent space at the identity of a Lie group will play a particularly important role as will be exposed in Section 4.

Example 2.11: Tangent space of Stiefel manifold

Similarly, for any $U \in St(k, n)$,

$$T_USt(k, n) = \{H \in M_{n,k}(\mathbb{R}) \mid U^\top H + H^\top U = 0\}.$$

We can now define the notion of smooth maps between manifolds and their differential.

Definition 2.7 (Smooth map)**.** Let M, Q be two manifolds of dimensions $d_1, d_2 \in \mathbb{N}$ embedded in $\mathbb{R}^N$. A function $f : M \to Q$ is *smooth* if for every $p \in M$, there are parametrizations $\varphi : V_1 \to U_1$ of M at p and $\psi : V_2 \to U_2$ of Q at $f(p)$ such that $f(U_1) \subseteq U_2$ and

$$\psi^{-1} \circ f \circ \varphi : V_1 \to V_2 \qquad \text{is smooth.}$$

Note that in the above definition V_1, V_2 are open sets respectively of $\mathbb{R}^{d_1}, \mathbb{R}^{d_2}$ so the notion of smooth map from V_1 to V_2 is already well known (see lexicon in Appendix A).

Definition 2.8 (Differential)**.** Let M, Q be two manifolds of dimensions $d_1, d_2 \in \mathbb{N}$ embedded in $\mathbb{R}^N$ and $f : M \to Q$ a smooth map. For any $p \in M$ and any $v \in T_pM$, let γ be a smooth curve through p such that $\dot\gamma(0) = v$ and define

$$df_p(v) = (f \circ \gamma)'(0).$$

This definition does not depend on the choice of curve γ and the map $df_p : T_pM \to T_{f(p)}Q$ is called the *differential* or *tangent map* of f at p. It is a linear map between tangent spaces.

It generalizes the differential map of a differentiable function defined from $\mathbb{R}^{N_1}$ to some $\mathbb{R}^{N_2}$ to functions defined on the manifold M instead of the embedding space. It coincides with the original differential (Definition A.11 in the lexicon page 96) when $M = \mathbb{R}^d$, hence the use of the same notation df_p. The set of real-valued smooth maps on M will be particularly useful. For short, we denote it $C^\infty(M) \triangleq C^\infty(M, \mathbb{R})$. $C^\infty(M)$ is clearly an infinite dimensional $\mathbb{R}$-vector space, and with point-wise multiplication, an algebra.

Next, it is convenient to consider the set of all the tangent spaces at all points

$$TM = \bigsqcup_{x \in M} \{x\} \times T_xM = \{(x, v) \mid x \in M, v \in T_xM\}.$$

and its natural projection

$$\pi : \left\{ \begin{array}{ccc} TM & \longrightarrow & M \\ (x, v) & \longmapsto & x \end{array} \right. .$$

This space is called the *tangent bundle* of M, and one can show that if M is a manifold of class C^{k+1} and dimension d, then TM is itself a manifold in $\mathbb{R}^N \times \mathbb{R}^N$, of class C^k and dimension $2d$. The tangent bundle is the domain of definition of the differential of smooth functions defined on manifolds:

$$f : M \to Q, \qquad df : TM \to TQ$$

It is also the space where *vector fields* are valued: a vector field X is a smooth assignment of a tangent vector to each point of a manifold, i.e. $X : M \to TM$ such that $\forall p \in M, \pi \circ X(p) = p$. $X(p)$ will be written X_p for convenience. Let $\Gamma(TM)$ denote the set of all vector fields (VF). It is clear that $\Gamma(TM)$ equipped with point-wise sum and multiplication by a scalar forms a vector space. Multiplication by a smooth function is also defined pointwise: for any $f \in C^\infty(M)$ and $X \in \Gamma(TM)$, fX is the vector field such that

$$\forall p \in M, (fX)_p = f(p)X_p.$$

This turns the set of vector fields into a $C^\infty(M)$-module. For a smooth map $f : M \to \mathbb{R}$ and a vector field X, we write $X(f)$ the function defined at every p by

$$X(f)(p) = df_p(X_p).$$

This leads to the following remark.

Remark 2.1. *We defined vector fields as sections of the tangent bundle, i.e., maps $\sigma : M \to TM$ such that $\pi \circ \sigma = Id$. Alternatively, vector fields can be defined as derivations over the algebra $C^\infty(M)$ of smooth real valued functions. A derivation $X : C^\infty(M) \to C^\infty(M)$ is a linear map that satisfies the Leibniz rule:*

$$\forall f, g \in C^\infty(M), \quad X(fg) = fX(g) + X(f)g. \tag{2.1}$$

One can check that a vector field as defined above indeed defines a derivation. However, applying the "composition" of two vector fields to a function f is not a derivation because of second-order derivatives of f. This leads to the definition of the Lie bracket of vector fields.

Definition 2.9 (Lie bracket over $\Gamma(TM)$)**.** The Lie bracket of vector fields is defined as the map

$$[\cdot, \cdot] : \left\{ \begin{array}{ccc} \Gamma(TM) \times \Gamma(TM) & \longrightarrow & \Gamma(TM) \\ (X, Y) & \longmapsto & f \mapsto X(Y(f)) - Y(X(f)) \end{array} \right. \tag{2.2}$$

A useful tool to handle vector fields locally is to use a basis of T_pM for p in some open set U.

Definition 2.10 (Frame)**.** Let M be a d-dimensional manifold. For any open set $U \subseteq M$, a family of vector fields $(X_1, \dots, X_d)$ over U is called a *frame* over U if for every p in U, $\big(X_1(p), \dots, X_d(p)\big)$ is a basis of T_pM.

Any chart (U, φ) defines a local frame that corresponds to its local coordinates: define the i^{th} curve $\gamma_i : t \in \mathbb{R} \mapsto \varphi^{-1}(0, \dots, 0, t, 0, \dots, 0) \in U$ and $X_i(\gamma_i(t)) = \dot{\gamma}_i(t)$. The vector field X_i defined on U is often written $X_i = \frac{\partial}{\partial x^i}$ or simply ∂_i and corresponds to $d\varphi^{-1}(e_i)$ where $(e_1, \dots, e_d)$ is the canonical basis of $\mathbb{R}^d$. Then the family $(X_1, \dots, X_d)$ is a local frame over U.

Remark 2.2. *If a family $(X_1, \dots, X_d)$ is a frame over the whole manifold (i.e. $U = M$), we say that it is a global frame. Whether global frames exist depends on the topology of the manifold, and in that case the tangent bundle is called trivial, i.e. isomorphic to the direct product $M \times \mathbb{R}^N$. This is not the case of e.g. the sphere (of dimension 2), by the hairy ball theorem* [4].

Vector fields can be considered as infinitesimal generators of local maps, as we shall see in the following. These maps, called flows, usually supply strong information on global properties of the manifold. In this monograph, we will mainly focus on geodesic flows (sec. 3.1), and flows of left-invariant vector fields on Lie groups (sec. 4.2).

Definition 2.11 (Integral curve)**.** Let $X \in \Gamma(M)$ and $p_0 \in M$. An *integral curve* for X with initial condition p_0 is a curve $\gamma : I \to M$ such that

$$\forall t \in I, \quad \dot{\gamma}(t) = X_{\gamma(t)} \quad \text{and} \quad \gamma(0) = p_0,$$

where $I = (a, b) \subseteq \mathbb{R}$ is an open interval containing 0.

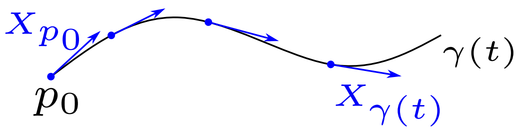


Figure 7: Example of an integral curve. The curve γ is obtained by integrating the vector field X (blue), meaning that $X_{\gamma(t)}$ is tangent to γ at all times.

An integral curve is thus a curve whose speed $\dot{\gamma}(t)$ coincide with X at any point along the curve (see Figure 7). A collection of such integral curves is called a flow (Figure 8):

Definition 2.12 (Local flow)**.** Let $X \in \Gamma(M)$ and $p_0 \in M$. A *local flow* of X at p_0 is a map $\phi : I \times U \to M$ where I is an open interval containing 0 and U is an open set in M containing p, such that for every $p \in U$, the curve $t \mapsto \phi(t, p)$ is an integral curve of X starting from p.

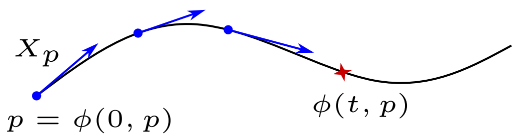


Figure 8: Example of a flow. The point $\phi(t, p)$ is reached at time t by the integral curve of the vector field X.

[4] https://en.wikipedia.org/wiki/Hairy_ball_theorem

Thanks to the theory of ordinary differential equations (ODE), one can prove that for any vector field, there is a local flow defined around any point, and if two such flows are defined on overlapping domains, they coincide on the intersection. For $t \in I$, we write $\phi_t : x \mapsto \phi(t, x)$. It is clear that ϕ_0 is the identity map, and for some $t \neq 0$, ϕ_t is a map defined locally on M. See Lafontaine, Gallot, et al. 2004, Proposition 1.55-56-58 for proofs.

Proposition 2.1. *Let X be a smooth vector field on M, $p_0 \in M$ and $\phi : I \times U \to M$ the local flow of X at p_0. For any $s, t \in I$ and $x \in U$,*

- *if $\phi_s(x) \in U$ and $t + s \in I$, then $\phi_t \circ \phi_s(x) = \phi_{t+s}(x)$;*
- *ϕ_t is a local diffeomorphism;*
- *ϕ_t preserves X, in the sense that $\forall t \in I, \forall x \in U$, $d(\phi_t)_x(X_x) = X_{\phi_t(x)}$.*

Definition 2.13 (Complete)**.** We say that X is *complete* if the domain of definition of its flow $(t, x) \mapsto \phi_t(x)$ is the entire $\mathbb{R} \times M$.

In that case ϕ_t is a diffeomorphism of M, and $(\phi_t)_{t\in\mathbb{R}}$ is a *one parameter subgroup* of $\mathrm{Diff}(M)$.

We now have the ingredients to introduce geomstats.

2.2 Implementation in geomstats

Now that the fundamental notion of manifold has been exposed, we can delve more into the architecture of the geomstats package, and summarize the choices that we made in its development. Firstly the package is organized in different modules that distinguish between the geometric and the statistical operations. There is thus a geometry module, that gathers all the implementations of manifolds, connections and Riemannian metrics, and a learning module where estimation algorithms are implemented in a generic fashion and take the geometric structure and the data as inputs. The goal is that all the learning algorithms can be run seamlessly on different manifolds, and with different metrics. Basic sampling schemes are available, and a more extended sampling module is currently being developed with the same spirit. A visualization module allows to plot data on the common manifolds in dimension two or three. In this section, we focus on the geometry module, and more specifically on the objects that represent manifolds. The Riemannian metric objects will be described in Section 3.1 along with the mathematical definitions. The statistical and learning tools will be described in Section 6.

The package is object-oriented in the sense that all the tools are implemented as classes that contain all the methods related to a tool. Object-Oriented programming (OOP) is a programming paradigm that consists in grouping properties and functions related to a common concept into an object, called a `class` in Python[5]

The geometry module of geomstats is organized by classes that each represent a geometric structure. To guarantee the consistency of all the classes, we implement an abstract parent `Manifold` class, and the actual implementations of the usual manifolds are subclasses of this parent class. The aim of abstract base classes is to provide the minimal skeleton of attributes and methods expected in its subclasses. The methods of the abstract class are thus declared but contain no implementation, and they are overridden by the subclasses.

The manifold classes A subclass of `Manifold` gathers the methods to work with data lying on the considered manifold. Note that these classes do not explicitly provide a representation of the manifold (as e.g. a triangulated surface in 2d), but the tools to handle

[5] For more on OOP, we refer the reader to online tutorials such as the one written by the RealPython team.

points and tangent vectors. As we work with embedded manifolds in most cases, points and tangent vectors are themselves represented by multi-dimensional arrays. These are either `NumPy` arrays, or `TensorFlow` of `PyTorch` tensors according to the backend that is being used.

Mathematically, the first attribute of a manifold is of course its dimension, called `dim` for brevity throughout the package. Then we use an attribute to inform on the expected type of the point: whether vectors (for e.g. the hypersphere and hyperbolic space) or matrices (e.g. SPD matrices, the special orthogonal group) should be used. This is called `default_point_type`.

Furthermore, a `Manifold` in geomstats should always implement a method that evaluates whether a given element belongs to that manifold, and whether a given vector is a tangent vector at a given point. These are the `belongs` and `is_tangent` methods. For practical reasons we also add a `random_point` method, to generate random points that belong to the manifold (regardless of the distribution). This is useful in particular to test the methods and the learning algorithms. We obtain the following base class:

```
class Manifold(abc.ABC):
    """Class for manifolds."""

    def __init__(
            self, dim, metric=None, default_point_type='vector',  **kwargs):
        super().__init__(**kwargs)
        self.dim = dim
        self.default_point_type = default_point_type
        self.metric = metric

    @abc.abstractmethod
    def belongs(self, point, atol=gs.atol):
        """Evaluate if a point belongs to the manifold."""

    @abc.abstractmethod
    def is_tangent(self, vector, base_point, atol=gs.atol):
        """Check whether the vector is tangent at base_point."""

    @abc.abstractmethod
    def random_point(self, n_samples=1, bound=1.):
        """Sample random points on the manifold."""
```

Note that the `Manifold` class contains a metric attribute. This will be detailed in Section 3.1.

Implementation trick 2.1. *The methods decorated (@ symbol) with `abc.abstractmethod` are declared as abstract methods of the class. A class that contains abstract methods cannot be instantiated. This constrains the developer to implement these functions explicitly when writing subclasses of `Manifold`.*

Two elementary classes of manifolds Throughout the current section, we have met two elementary ways of defining a manifold, that correspond respectively to (1) and (2) of Theorem 2.1 (page 7):

1. As pre-image of a submersion $f : \mathbb{R}^N \to \mathbb{R}^{N-d}$. We refer to such space as a level-set. In this case, specifying f, it is straightforward to implement the `belongs` and `is_tangent` method by evaluating f and its differential. It also makes sense to add `projection` and `to_tangent` methods from the embedding space to respectively the manifold and the tangent space at a point.

2. As open sets of a d-dimensional vector space, called ambient space. In this case, all the tangent spaces are identified with the ambient space. For consistency, we add a `projection` method that maps any d-dimensional vector to the manifold at a tolerance threshold $\epsilon > 0$ away from the boundary of the set (if there is one).

This method is not uniquely defined and can be understood as a regularization method for inputs very close to the open set, which is helpful in learning algorithms. Therefore, the method `is_tangent` just checks if the input belongs to the ambient space, and we can add a method `to_tangent`, that calls the projection of the ambient space, to project to a tangent space, assuming the ambient space to be itself embedded in another space, or being a vector space, whose projection is the identity.

We thus implement two more abstract classes, the first for open sets:

```
class OpenSet(Manifold, abc.ABC):
    """Class for manifolds that are open sets of a vector space."""

    def __init__(self, dim, ambient_space, **kwargs):
        kwargs.setdefault("shape", ambient_space.shape)
        super().__init__(dim=dim, **kwargs)
        self.ambient_space = ambient_space

    def is_tangent(self, vector, base_point, atol=gs.atol):
        """Check whether the vector is tangent at base_point."""
        return self.ambient_space.belongs(vector, atol)

    def to_tangent(self, vector, base_point):
        """Project a vector to a tangent space of the manifold."""
        return self.ambient_space.projection(vector)

    def random_point(self, n_samples=1, bound=1.):
        """Sample random points on the manifold."""
        sample = self.ambient_space.random_point(n_samples, bound)
        return self.projection(sample)

    @abc.abstractmethod
    def projection(self, point):
        """Project a point in ambient manifold on manifold."""
```

And the second for level-sets:

```
class LevelSet(Manifold, abc.ABC):
    """Class for manifolds embedded in a vector space by a submersion."""

    def __init__(
            self, dim, embedding_space, submersion, value,
            tangent_submersion, **kwargs):
        super().__init__(
            dim=dim, default_point_type=embedding_space.default_point_type,
            **kwargs)
        self.embedding_space = embedding_space
        self.embedding_metric = embedding_space.metric
        self.submersion = submersion
        self.value = value
        self.tangent_submersion = tangent_submersion

    def belongs(self, point, atol=gs.atol):
        """Evaluate if a point belongs to the manifold."""
        belongs = self.embedding_space.belongs(point, atol)
        if not gs.any(belongs):
            return belongs
        value = self.value
        constraint = gs.isclose(self.submersion(point), value, atol=atol)
        if value.ndim == 2:
            constraint = gs.all(constraint, axis=(-2, -1))
        elif value.ndim == 1:
            constraint = gs.all(constraint, axis=-1)
        return gs.logical_and(belongs, constraint)
```

```python
    def is_tangent(self, vector, base_point, atol=gs.atol):
        """Check whether the vector is tangent at base_point."""
        belongs = self.embedding_space.belongs(vector, atol)
        tangent_sub_applied = self.tangent_submersion(vector, base_point)
        constraint = gs.isclose(tangent_sub_applied, 0., atol=atol)
        value = self.value
        if value.ndim == 2:
            constraint = gs.all(constraint, axis=(-2, -1))
        elif value.ndim == 1:
            constraint = gs.all(constraint, axis=-1)
        return gs.logical_and(belongs, constraint)

    @abc.abstractmethod
    def projection(self, point):
        """Project a point in embedding space on the manifold."""
```

To make sure that the attributes that represent the ambient/embedding space do implement the methods that are called in `OpenSet` and `LevelSet`, we also implemented an abstract `VectorSpace` class. Actual manifolds are then implemented as subclasses of the corresponding abstract manifold and must implement all the abstract methods.

To illustrate, a diagram representing all the base classes for manifolds and all the manifolds is shown on Figure 9. The abstract base classes are shown with black bounding boxes. Inheritances occur in two cases:

- With an abstract base class as parent class;
- When both parent and child class represent the same manifold, as is the case for example in the `CorrelationBundle` of Example 5.8. A common interface `Hyperbolic` for the three representations of hyperbolic geometry follows this logic as any of the three representations can be chosen, but the instantiated object is either `Hyperboloid`, `PoincareBall` or `PoincareHalfSpace` as chosen by the user.

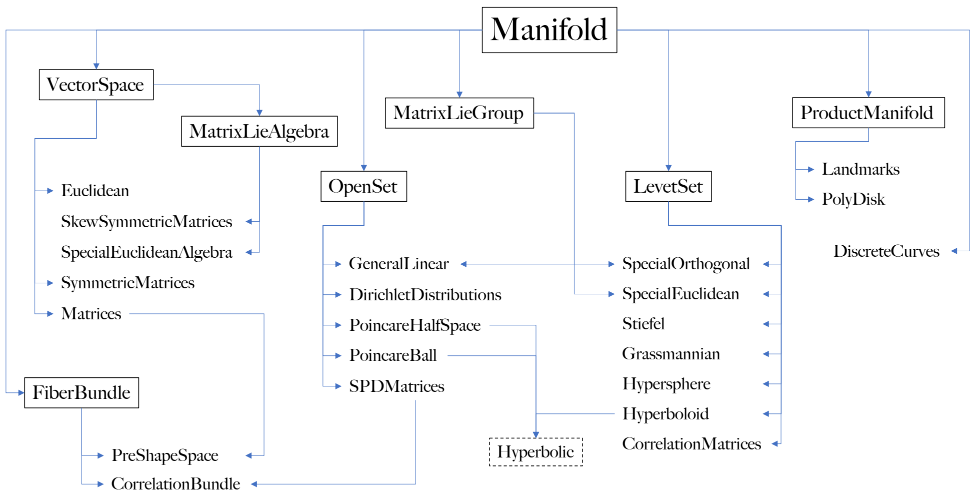


Figure 9: Architecture of the manifolds of geomstats. The abstract base classes are in black bounded boxes. Inheritance is shown by blue arrows. `Hyperbolic` is an exception as it is a common interface to the different representations of hyperbolic geometry.

Implementation trick 2.2. *All the methods of geomstats are vectorized*[6]*, in the sense that they can take as argument either one input, or a collection of inputs corresponding*

[6] Vectorization may also be referred to as *array programming* on e.g. Wikipedia.

to multiple samples. It is very useful to use the `einsum` *method for that purpose, with ellipses ('...') that represent an optional additional dimension.*

For example, the syntax `gs.einsum('...,...i->...i', coef, point)` *performs scalar multiplication between a list of scalars (*`coef`*) and a list of* `points`*, but it also works for a single scalar and a single point.*

We give examples of each class below.

Example 2.12: Implementation of the hypersphere

The hypersphere is implemented as embedded in $\mathbb{R}^{d+1}$, so it is a subclass of `LevelSet`.

```
class Hypersphere(LevelSet):
    """Class for the n-dimensional hypersphere."""

    def __init__(self, dim):
        super().__init__(
            dim=dim, embedding_space=Euclidean(dim + 1),
            metric=HypersphereMetric(dim),
            submersion=lambda x: gs.sum(x ** 2, axis=-1), value=1.,
            tangent_submersion=lambda v, x: 2 * gs.sum(x * v, axis=-1))

    def projection(self, point):
        """Project a point on the hypersphere."""
        norm = gs.linalg.norm(point, axis=-1)
        if gs.any(norm < gs.atol):
            logging.warning('0 cannot be projected to the hypersphere')
        return gs.einsum('...,...i->...i', 1. / norm, point)

    def to_tangent(self, vector, base_point):
        """Project a vector to the tangent space."""
        sq_norm = gs.sum(base_point ** 2, axis=-1)
        inner_prod = self.embedding_metric.inner_product(base_point, vector)
        coef = inner_prod / sq_norm
        return vector - gs.einsum('...,...j->...j', coef, base_point)
```

Example 2.13: Implementation of the Stiefel manifold

The Stiefel manifold is implemented as embedded in the space of $n \times p$ matrices, so it is a subclass of `LevelSet`. The derivation of the projection map can be found in Absil and Malick 2012.

```python
class Stiefel(LevelSet):
    """Class for Stiefel manifolds St(n,p."""

    def __init__(self, n, p, **kwargs):
        if p > n:
            raise ValueError("p needs to be smaller than n.")

        dim = int(p * n - (p * (p + 1) / 2))
        matrices = Matrices(n, p)
        canonical_metric = StiefelCanonicalMetric(n, p)
        kwargs.setdefault("metric", canonical_metric)
        super(Stiefel, self).__init__(
            dim=dim,
            embedding_space=matrices,
            submersion=lambda x: matrices.mul(matrices.transpose(x), x),
            value=gs.eye(p),
            tangent_submersion=lambda v, x: 2
            * matrices.to_symmetric(matrices.mul(matrices.transpose(x), v)),
            **kwargs
        )
        self.n = n
        self.p = p

    def to_tangent(self, vector, base_point):
        """Project a vector to a tangent space of the manifold."""
        aux = Matrices.mul(Matrices.transpose(base_point), vector)
        sym_aux = Matrices.to_symmetric(aux)
        return vector - Matrices.mul(base_point, sym_aux)

    def projection(self, point):
        """Project a close enough matrix to the Stiefel manifold."""
        mat_u, _, mat_v = gs.linalg.svd(point)
        return Matrices.mul(mat_u[..., :, : self.p], mat_v)
```

Example 2.14: Implementation of the Poincaré ball

The Poincare ball is one of the models of hyperbolic geometry, and is defined as the open unit disk of $\mathbb{R}^d$. It is then a subclass of `OpenSet`.

```python
class PoincareBall(OpenSet):
    """Class for the n-dimensional hyperbolic space."""

    def __init__(self, dim, scale=1):
        super().__init__(
            dim=dim, ambient_space=Euclidean(dim),
            metric=PoincareBallMetric(dim))

    def belongs(self, point, atol=gs.atol):
        """Test if a point belongs to the unit ball."""
        return gs.sum(point**2, axis=-1) < (1 - atol)

    def projection(self, point):
        """Project a point on the unit ball."""
        if point.shape[-1] != self.dim:
            raise NameError("Wrong dimension, expected ", self.dim)

        l2_norm = gs.linalg.norm(point, axis=-1)
        if gs.any(l2_norm >= 1 - gs.atol):
            projected_point = gs.einsum(
                '...j,...->...j', point * (1 - gs.atol), 1. / l2_norm)
            projected_point = -gs.maximum(-projected_point, -point)
            return projected_point
        return point
```

Manifolds can then be composed to define other manifolds by products (Example 2.6) or quotients (see Section 5.1.4). For products, we create the class `ProductManifold` that takes existing manifolds to construct a new one, and computations on each manifold can be done in parallel:

```
class ProductManifold(Manifold):
    r"""Class for a product of manifolds M_1 \times ... \times M_n."""

    def __init__(self, manifolds, n_jobs=1):
        self.dims = [manifold.dim for manifold in manifolds]
        super().__init__(dim=sum(self.dims))
```

3 Riemannian manifolds

3.1 Riemannian metrics

We now introduce a new structure on a differentiable manifold: the Riemannian metric, that allows to define the length of a curve, a distance function, a volume form, etc. Note that this additional structure may not be canonical, raising the thorny question of choosing the metric for the applications.

Recall that TM is the tangent bundle of a smooth manifold M (defined page 14).

Definition 3.1 (Riemannian metric)**.** Let M be a smooth d-dimensional manifold. A *Riemannian metric* on M (or TM) is a family $(\langle\cdot,\cdot\rangle_p)_{p\in M}$ of inner products* on each tangent space T_pM, such that $\langle\cdot,\cdot\rangle_p$ depends smoothly on p. More formally, for any chart φ, U, and frame $(X_1,\dots,X_n)$ on U, the maps

$$p \mapsto \langle X_i(p), X_j(p)\rangle_p \quad 1 \leq i,j \leq n,$$

are smooth. A pair $(M, \langle\cdot,\cdot\rangle)$ is called a *Riemannian manifold*.

A metric is often written $g = (g_p)_{p\in M}$, where g_p is the symmetric, positive definite (SPD) matrix representing the inner-product in a chart, that is

$$g_{ij}(p) = \left\langle (\partial_i)_p, (\partial_j)_p \right\rangle_p.$$

Alternatively, we sometimes define a metric with the notation $g = f(dx_1,\dots,dx_n)$ where f is the quadratic form associated with g, and dx_i represent vector coordinates (as linear forms). For example the usual Euclidean metric is $g = \sum_{i=1}^n dx_i^2$. The following theorem ensures that a metric is indeed a general structure. A proof can be found e.g. in Lafontaine, Gallot, et al. 2004, Theorem 2.2.

Theorem 3.1 (Existence)**.** *Any smooth manifold admits a Riemannian metric.*

A Riemannian metric also defines a norm on TM, as usually defined by an inner-product on each tangent space:

$$\forall x \in M, \forall v \in T_xM, \quad \|v\|_x = \sqrt{g_x(v,v)} = \sqrt{\langle v,v\rangle_x}.$$

Implementation in geomstats In order to guarantee flexibility, we have decided to keep manifolds and metrics separated in different objects. Indeed, although there may exist a canonical Riemannian metric on a given manifold, the choice of metric is not always natural, and researchers have struggled to find criteria to choose the right metric for the application at hand. The aim of geomstats is to allow researchers to compare

*Defined in Appendix A.

different metrics on their problem, and in the future, to allow to *learn*, or optimize the metric (Louis, Couronné, et al. 2019; Hauberg 2019).

We create an abstract `RiemannianMetric` class in geomstats to gather the basic attributes and methods expected of a metric. The most general way of defining a metric is to provide the `metric_matrix` method, that is $x \mapsto (g_{ij}(x))_{1 \leq i,j \leq d}$. By default, we use the identity matrix for all points, resulting in the Euclidean metric.

```
class RiemannianMetric(Connection):
    """Class for Riemannian and pseudo-Riemannian metrics."""

    def __init__(self, dim, signature=None):
        super().__init__(dim=dim)
        if signature is None:
            self.signature = (dim, 0)

    def metric_matrix(self, base_point=None):
        """Inner product matrix at base point."""
        return gs.eye(self.dim)

    def inner_product(self, tangent_vec_a, tangent_vec_b, base_point=None):
        """Inner product between two tangent vectors at a base point."""
        inner_prod_mat = self.metric_matrix(base_point)
        inner_prod = gs.einsum(
            '...j,...jk,...k->...', tangent_vec_a, inner_prod_mat, tangent_vec_b)
        return inner_prod

    def squared_norm(self, vector, base_point=None):
        """Compute the square of the norm of a vector."""
        return self.inner_product(vector, vector, base_point)

    def norm(self, vector, base_point=None):
        """Compute norm of a vector."""
        sq_norm = self.squared_norm(vector, base_point)
        return gs.sqrt(sq_norm)
```

Remark 3.1.

1. *The above class inherits from* `Connection`*. We indeed chose a class for an affine connection as parent class to a Riemannian metric because it is a more general structure, as explained in paragraph 3.2.1.*

2. *The attribute* `signature` *refers to the signature of the inner product, in case it is only a non-degenerate bilinear form, not necessarily positive. In this case the metric is called a pseudo-Riemannian metric.*

We also add a `metric` property to the `Manifold` class, meaning that it is an attribute of the class, that can be set externally, calling the `setter` that checks that the given argument is indeed an instance of a `RiemannianMetric` object. Of course, all manifolds studied in this monograph come with a default metric, but users can choose to use different metrics or implement new metrics with only a few minimal operations. When closed form solutions are available, the generic methods are overridden. With a metric attribute, a `Manifold` actually becomes a Riemannian manifold, but we did not think relevant to implement another layer of abstract class for Riemannian manifolds (i.e. a `RiemannianManifold` class), as all necessary operations are either in the `Manifold` object if they don't depend on the metric, or in the `RiemannianMetric` object if they do.

```
    @property
    def metric(self):
        """Riemannian Metric associated to the Manifold."""
        return self._metric
```

```
    @metric.setter
    def metric(self, metric):
        if metric is not None:
            if not isinstance(metric, RiemannianMetric):
                raise ValueError(
                    'The argument must be a RiemannianMetric object')
            if metric.dim != self.dim:
                metric.dim = self.dim
        self._metric = metric
```

Example 3.1: Euclidean metric

Let $M = \mathbb{R}^d$ be the standard vector space of dimension d, that is trivially a smooth manifold, and consider its standard inner-product defined for all $x, y \in \mathbb{R}^d$ by

$$\langle x, y\rangle_2 = \sum_{i=1}^{d} x_i y_i = x^\top y.$$

As $T_x\mathbb{R}^d = \mathbb{R}^d$, it defines a metric on $\mathbb{R}^d$, which is referred to as the Euclidean metric.

Example 3.2: Product metric

Let (M, g) and (M', g') be two Riemannian manifolds, and recall from Example 2.6 (page 10) that the Cartesian product $M \times M'$ is a manifold. It is also a Riemannian manifold. Indeed, define the *product metric* $g \oplus g'$ as the map defined at any $(x, x') \in M \times M'$ and $\forall (v, v'), (w, w') \in T_xM \times T_{x'}M'$ by

$$g \oplus g'_{(x,x')}((v, v'), (w, w')) = g(v, w) + g'(v', w').$$

In geomstats, it is possible to define such product metrics from existing objects of the class `RiemannianMetric`, and operations for each metric can be run in parallel if necessary.

```
class ProductRiemannianMetric(RiemannianMetric):
    """Class for product of Riemannian metrics."""

    def __init__(self, metrics, default_point_type='vector', n_jobs=1):
        self.n_metrics = len(metrics)
        dims = [metric.dim for metric in metrics]
        signatures = [metric.signature for metric in metrics]

        sig_pos = sum(sig[0] for sig in signatures)
        sig_neg = sum(sig[1] for sig in signatures)
        super().__init__(
            dim=sum(dims), signature=(sig_pos, sig_neg),
            default_point_type=default_point_type)
```

Let (N, g) be a Riemannian manifold, M a smooth manifold, and $f : M \to N$ a map. Define the *pull-back* metric (f^*g) on M by

$$(f^*g)_x : \left\{ \begin{array}{ccc} T_xM \times T_xM & \longrightarrow & \mathbb{R} \\ (v, w) & \longmapsto & g_{f(x)}(df_xv, df_xw) \end{array} \right. . \tag{3.1}$$

Now suppose that f is an immersion. If $(f^*g)_x$ is non-degenerate and of constant signature for all $x \in M$, then (M, f^*g) is a Riemannian manifold.

Definition 3.2 (Isometry)**.** Let (M, g) and (M', g') be two Riemannian manifolds, and $f : M \to M'$. Then f is called an *isometry* if it is a bijection and $f^* g' = g$.

When $M = M'$ and $g = g'$, we write $\mathrm{Isom}(M)$ the set of isometries of M, and Myers and Steenrod 1939 showed that it is a Lie group that acts smoothly on M.

Now, consider that $M \subseteq N = \mathbb{R}^d$ is a submanifold of M, $f = i$ is the inclusion map. Then g is called the embedding metric and $i^* g$ is its restriction to M. This case appears in many examples in geomstats.

Example 3.3: Metric on the hypersphere

The hypersphere $S^d \subset \mathbb{R}^{d+1}$ endowed with the restriction of the Euclidean metric to S^d is a Riemannian manifold. We call this metric the standard spherical metric, and implement it by calling the embedding metric.

```
class HypersphereMetric(RiemannianMetric):
    """Class for the Metric on the Hypersphere."""

    def __init__(self, dim):
        super().__init__(
            dim=dim, signature=(dim, 0))
        self.embedding_metric = EuclideanMetric(dim + 1)

    def metric_matrix(self, base_point=None):
        """Inner-product matrix at a base point."""
        return gs.eye(self.dim + 1)

    def inner_product(self, tangent_vec_a, tangent_vec_b, base_point=None):
        """Inner-product of two tangent vectors at a base point."""
        return self.embedding_metric.inner_product(
            tangent_vec_a, tangent_vec_b, base_point)
```

let us now define the Lorentz bilinear form of $\mathbb{R}^{d+1}$. It is the canonical bilinear form with signature (1, d), i.e. for any $x, y \in \mathbb{R}^{d+1}$

$$\langle x, y \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^{d} x_i y_i \tag{3.2}$$

And write $\| \cdot \|_{\mathcal{L}}$ for the associated quadratic form. This is the underlying embedding metric in the following example.

Example 3.4: Hyperbolic metric

Using the Lorentz form, recall that $H^d \in \mathbb{R}^{d+1}$ is the set of points such that $\|x\|_{\mathcal{L}} = -1$. Consider now the open subset H_+^d of H^d:

$$H_+^d = \left\{ x \in \mathbb{R}^{d+1} \mid x_0 > 0, \ \|x\|_{\mathcal{L}} = -1 \right\}.$$

Now, consider any $x \in H_+^d$ and two tangent vectors $v, w \in T_x M$. By Example 2.9, this means that $\langle v, x \rangle_{\mathcal{L}} = \langle w, x \rangle_{\mathcal{L}} = 0$. As $\langle , \rangle_{\mathcal{L}}$ is negative definite on $\mathbb{R}x = \{\lambda x \mid \lambda \in \mathbb{R}\}$, and its signature is $(1, d)$, it is positive definite on the orthogonal of $\mathbb{R}x$ for the Lorentz metric, i.e. on $T_x M$. We can thus conclude that $(H_+^d, \langle \cdot, \cdot \rangle_{\mathcal{L}})$ is a Riemannian manifold.

In the implementation, we use the abstract class and simply override the `inner_product` function.

```python
class HyperbolicMetric(RiemannianMetric):
    """Class for the hyperbolic metric."""

    def __init__(self, dim):
        super().__init__(
            dim=dim, signature=(1, dim))

    def metric_matrix(self, base_point=None):
        """Inner product matrix at base point."""
        diagonal = gs.array([-1.] + [1.] * self.dim)
        return from_vector_to_diagonal_matrix(diagonal)

    def inner_product(self, tangent_vec_a, tangent_vec_b, base_point=None):
        """Inner product between two tangent vectors at a base point."""
        diagonal = gs.array([-1.] + [1.] * self.dim)
        return gs.sum(diagonal * tangent_vec_a, tangent_vec_b, axis=-1)
```

Example 3.5: Frobenius metric

The analog of the Euclidean metric on matrix spaces is the *Frobenius* inner product defined by

$$\forall A, B \in M_{m,n}(\mathbb{R}), \quad \langle A, B \rangle_F = \mathrm{tr}(A^\top B) = \sum_{i,j} A_{ij} B_{ij}.$$

where tr is the trace operator. We use the right-hand-side expression in geomstats to avoid computing a matrix product ($O(mn^2)$ operations against $O(mn)$).

Endowed with this metric (or its restriction), $M_n(\mathbb{R})$, $GL(n)$ (Example 4.1 page 40), $SO(n)$ (Example 2.3 page 8) and $SE(n)$ (Example 4.7 page 45) are Riemannian manifolds.

3.2 Affine connections and the Levi-Civita connection

A connection is an additional structure that can be defined independently of a Riemannian metric. It provides a way to compare tangent spaces from one point to another, by defining the notion of parallelism. For a detailed and historical account of the different approaches to defining connections, we refer to Marle 2005.

3.2.1 Connections

Definition 3.3 (Connection)**.** Let M be a smooth manifold. A *connection* on M is an $\mathbb{R}$-bilinear map $\nabla : \Gamma(TM) \times \Gamma(TM) \to \Gamma(TM)$ that verifies for all $X, Y \in \Gamma(TM), \forall f \in C^\infty(M)$:

1. (Linearity of 1^{st} argument) $\nabla_{fX} Y = f \nabla_X Y$,
2. (Leibniz rule in 2^{nd} argument) $\nabla_X(fY) = X(f)Y + f\nabla_X Y$.

The vector field $\nabla_X Y$ is called the *covariant derivative* of Y w.r.t. X.

In fact, $(\nabla_X Y)_p$ only depends on the value of X at p and not in its neighborhood. In contrast, it does depend on Y around p.

In local coordinates $(x^1, \ldots, x^d)$, the *Christoffel symbols* are used to specify the connection. Recall that $\big(\partial_i\big)_i$ is a local frame, so we can decompose ∇ in this basis, and define $(\Gamma^k_{ij})_{ijk}$ such that

$$\nabla_{\partial_i}\big(\partial_j\big) = \Gamma^k_{ij} \partial_k,$$

where we used Einstein summation convention, meaning that a sum occurs along the indices that appear both in subscript and superscript, here k.

Two vector fields $X, Y \in \Gamma(TM)$ can be decomposed locally in coordinates, writing $X = X^i\partial_i$ and $Y = Y^i\partial_i$ where X^i, Y^i are coordinate functions defined locally. Then using the properties of a connection,

$$\begin{aligned}\nabla_X Y &= X^i \nabla_{\partial_i}\big(Y^j\partial_j\big)\\ \nabla_X Y &= X^i\big(\frac{\partial Y^j}{\partial x^i}\partial_j + Y^j\Gamma^k_{ij}\partial_j\big).\end{aligned} \tag{3.3}$$

As this formula shows, a connection provides a correction term when compared to the directional derivative of Y with respect to X in a chart.

3.2.2 Parallel transport and geodesics

We now focus on vector fields that are defined along a curve $\gamma : [a, b] \to M$, i.e. a smooth map $X : [a, b] \to TM$ such that at any $t \in [a, b]$, $X(t) \in T_{\gamma(t)}M$. Note that such vector field need not be defined on the whole manifold, but can be locally extended to an open set around every point. Thankfully, one can show that there exists a *covariant derivative* that coincides with $(\nabla_{\dot\gamma(t)}X)_{\gamma(t)}$ at all $t \in [a, b]$ and for any X defined on a neighborhood of $\gamma(t)$ (Lafontaine, Gallot, et al. 2004, Theorem 2.68). We will skip these technicalities and admit that $\nabla_{\dot\gamma}X$ is well-defined for any vector field X along γ.

We now define a central notion in geometric statistics: the parallel transport.

Definition 3.4 (Parallel vector field)**.** Let M be a smooth manifold and ∇ a connection on M. For any curve $\gamma : [a, b] \to M$ in M, a vector field X along γ is *parallel* if

$$\nabla_{\dot\gamma(t)}X(t) = 0. \tag{3.4}$$

In a local chart and using the Christoffel symbols and in particular using eq. (3.3), eq. (3.4) can be written $\forall t \in [a, b]$

$$\dot X^k(t) + \Gamma^k_{ij}X^i(t)\dot\gamma^j(t) = 0. \tag{3.5}$$

From the properties of ODEs, one can prove the following existence and uniqueness property (Lafontaine, Gallot, et al. 2004, Proposition 2.72)

Proposition 3.1. *Let M be a smooth manifold and let ∇ be a connection on M. For every C^1 curve $\gamma : [a, b] \to M$ in M, for every $t \in [a, b]$ and every $v \in T_{\gamma(t)}M$, there is a unique parallel vector field X along γ such that $X(t) = v$.*

For such parallel vector field X and $s \in [a, b]$, we thus call $X(s)$ the *parallel transport* of v along γ from t to s, and write $X(s) = \Pi^s_{\gamma,t}v$. Another consequence of the properties of ODEs is that for any $s, t \in [a.b]$, $\Pi^s_{\gamma,t}$ is a linear isomorphism between the tangent spaces $T_{\gamma(t)}M$ and $T_{\gamma(s)}M$.

Intuitively, the parallel transport equation constrains the transported vector to keep a certain angle w.r.t the speed $\dot\gamma$ of the curve while moving along it, see Figure 10 for an illustration. Conversely, the connection $\nabla_X Y$ could be retrieved from the parallel transport $\Pi^s_{\gamma,t}Y$ when the curve is shrunk to a point in the direction of X, i.e. $s \to t$ and $\dot\gamma(t) = X$.

We now focus on a particular set of curves on M for which the velocity is parallel transported along the curve.

Definition 3.5. Let M be a smooth manifold endowed with a connection ∇. A curve $\gamma : [a, b] \to M$ is said to be *autoparallel* and is called a *geodesic* of (M, ∇) if it satisfies for all $t \in [a, b]$

$$\nabla_{\dot\gamma(t)}\dot\gamma(t) = 0. \tag{3.6}$$

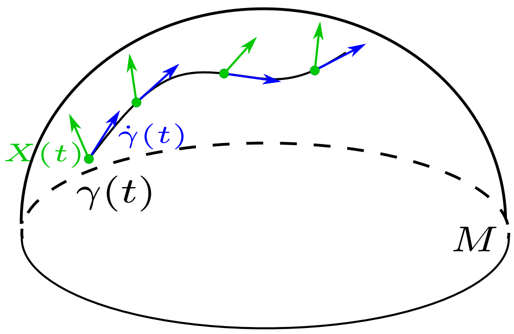


Figure 10: Representation of a parallel vector field X (green) along a curve γ (blue). The orientation of X with resepct to the speed $\dot\gamma$ of γ stays the same.

Note that $\nabla_{\dot\gamma}\dot\gamma$ can be interpreted as the covariant acceleration of γ, so that equation (3.6) constrains geodesics to be zero acceleration curves. The flow of the geodesic equation is called geodesic flow, and is a fundamental example of dynamical system, that generalizes straight lines from Euclidean spaces.

The geodesic equation can be written in local coordinates like the parallel transport equation. A geodesic curve γ satisfies for all times $t \in [a, b]$

$$\ddot\gamma^k(t) + \Gamma^k_{ij}\dot\gamma^i(t)\dot\gamma^j(t) = 0. \tag{3.7}$$

From the properties of second-order differential equations, for any $(x, v) \in TM$, there exists a maximal interval $I_{x,v} \subseteq \mathbb{R}$ such that $\gamma_{x,v} : I_{x,v} \to M$ is the unique geodesic that verifies $\gamma_{x,v}(0) = x$ and $\dot\gamma_{x,v}(0) = v$. Moreover, by homogeneity of the equation (3.7), for any $s > 0$, $I_{x,sv} = \frac{1}{s} I_{x,v}$ and $\gamma_{x,sv}(t) = \gamma_{x,v}(st)$. We deduce that the set of vectors in T_xM such that $1 \in I_{x,v}$ is non empty, open, and contains 0. This leads to the following definition

Definition 3.6 (Exponential map)**.** Let ∇ be a connection on a smooth manifold M. The map $(x, v) \mapsto \gamma_{x,v}(1)$ defined on the open set $\{(x, v) \in TM, 1 \in I_{x,v}\}$ and with values in M is called the *exponential map* of ∇. We say that M is *geodesically complete* if the exponential map is defined on the whole of TM.

For any $x \in M$, we write the exponential map at x $\mathrm{Exp}_x : v \in T_xM \mapsto \gamma_{x,v}(1)$.

Remark 3.2. *Note that although we introduced many notions in this paragraph, there were no examples. Indeed a connection or its Christoffel symbols are rarely explicit, except when the connection is compatible with a metric. We explain this notion of compatibility in the next paragraph, and will give several examples of geodesics and parallel transports.*

3.2.3 The Levi-Civita connection

First, given two vector fields Y, Z, recall that $\langle Y, Z\rangle$ is a smooth function on M so $X(\langle Y, Z\rangle)$ must be understood as $X(f) = df(X)$ for $f : p \mapsto \langle Y_p, Z_p\rangle_p$.

The following is considered the fundamental theorem of Riemannian geometry. It ensures that there exists a unique connection that is "compatible" with the metric. See Lafontaine, Gallot, et al. 2004, Theorem 2.51 for a proof.

Theorem 3.2. *Let (M, g) be a Riemannian manifold. There is a unique connection on M that verifies for all $X, Y, Z \in \Gamma(TM)$*

1. *(Torsion-free)* $$\nabla_X Y - \nabla_Y X = [X, Y] \tag{3.8}$$
2. *(Compatibility)* $$X(\langle X, Y\rangle) = \langle \nabla_X Y, Z\rangle + \langle Y, \nabla_X Z\rangle \tag{3.9}$$

This connection is called the Levi-Civita connection and is determined by the Koszul formula:

$$\begin{aligned} 2\langle \nabla_X Y, Z\rangle = X(\langle Y, Z\rangle) + Y(\langle X, Z\rangle) - Z(\langle X, Y\rangle) \\ - \langle Y, [X, Z]\rangle - \langle X, [Y, Z]\rangle - \langle Z, [Y, X]\rangle. \end{aligned} \tag{3.10}$$

The notion of compatibility is thus detailed in equations (3.8) and (3.9). The former is quite general but ensures uniqueness of the Levi-Civita connection, it is called the zero-torsion condition. The latter can be understood as a Leibniz rule where $\langle Y, Z\rangle$ is seen as a product and the derivative is ∇_X. More precisely, this condition means that the metric is parallel with respect to the connection. Indeed, although we introduced a connection as a map on vector fields, it can be extended to tensors of any order. For a 0-order tensor, i.e. a smooth function f, we have $\nabla_X f = X(f)$, and for a (2, 0)-tensor such as g, we have (by generalising the Leibniz rule) for any $X, Y, Z \in \Gamma(TM)$

$$\begin{aligned} (\nabla_X g)(Y, Z) &= \nabla_X(g(Y, Z)) - g(\nabla_X Y, Z) - g(Y, \nabla_X Z). \\ &= X(\langle Y, Z\rangle) - \langle \nabla_X Y, Z\rangle - \langle Y, \nabla_X Z\rangle \end{aligned} \tag{3.11}$$

Therefore the combination of equations (3.9) and (3.11) results in $\nabla g = 0$.

In the general case, the Levi-Civita connection is characterized locally by its Christoffel symbols. Let $(x^1, \ldots, x^n)$ be a local coordinate chart, by definition

$$\nabla_{\partial_i}\partial_j = \Gamma^k_{ij}\partial_k$$

The torsion-free condition (3.8) and Schwartz theorem ($[\partial_i, \partial_j] = 0$) thus imply the symmetry of the symbols: $\Gamma^k_{ij} = \Gamma^k_{ji}$ for all $1 \leq i, j, k \leq n$. Furthermore, together with (3.9), Koszul formula (3.10) translates into

$$2g(\nabla_{\partial_i}\partial_j, \partial_k) = \partial_i g_{jk} + \partial_j g_{ki} - \partial_k g_{ij}.$$

Writing $(g^{ij})_{ij} = (g_{ij})^{-1}_{ij}$ for the inverse of the metric matrix, we obtain

$$\Gamma^k_{ij} = \frac{1}{2} g^{lk}(\partial_i g_{jl} + \partial_j g_{li} - \partial_l g_{ij}) \tag{3.12}$$

Thus the Christoffel symbols can be computed from the metric g. This formula is rarely used by mathematicians for computations "by hand", but we shall use it in geomstats to implement pull back metrics.

The following characterization allows to compute the Levi-Civita connection in the case of embedded manifolds equipped with the embedding metric, as is the case in many of our examples. See Lafontaine, Gallot, et al. 2004, Proposition 2.56 for a proof.

Proposition 3.2. *Let M be a Riemannian manifold and N a submanifold of M endowed with the induced metric (i.e., the restriction of the embedding metric). If ∇^M and ∇^N are the Levi-Civita connections on M and N respectively defined by the metric on M. Then for any vector field $X, Y \in \Gamma(TN)$, we have*

$$\nabla^N_X Y = (\nabla^M_X Y)^{\parallel}, \tag{3.13}$$

where $(v)^{\parallel}$ is the orthogonal projection of any $v \in T_pM$ onto T_pN for any $p \in N$.

This proposition is particularly useful for embedded manifolds of $\mathbb{R}^N$ endowed with the ambient Euclidean metric, as it is straightforward to see (by uniqueness) that their Levi-Civita connection coincide with the directional derivative of functions from $\mathbb{R}^N$ to $\mathbb{R}^N$. Therefore, one only needs to compute the projection to the tangent spaces of any point of an embedded manifold to compute its connection.

Finally, thanks to the compatibility of the metric with the connection, the parallel transport preserves the metric, as stated in the following proposition (Lafontaine, Gallot, et al. 2004, Proposition 2.74).

Proposition 3.3. *Let $\gamma : I \to M$ be a smooth curve, and $s, t \in I$. Then the parallel transport map $\Pi_{\gamma,s}^t : T_{\gamma(s)}M \to T_{\gamma(t)}M$ along γ for the Levi-Civita connection is an isometry, i.e.*

$$\forall v, w \in T_{\gamma(s)}M, \quad \langle \Pi_{\gamma,s}^t v, \Pi_{\gamma,s}^t w\rangle_{\gamma(t)} = \langle v, w\rangle_{\gamma(s)}.$$

This justifies the use of parallel transport in statistical procedures to move data from one reference point to another while preserving distances.

Moreover, the compatibility with the metric also ensures that isometries preserve the Levi-Civita connection and its geodesics. The following can be deduced from the Kozsul formula (3.10).

Proposition 3.4. *Let (M, g) be a Riemannian connection and ∇ its Levi-Civita connection. Let $f \in \mathrm{Isom}(M)$ be an isometry of (M, g), then for any $X, Y \in \Gamma(TM)$ and $x \in M$, we have*

$$df \nabla_X Y = \nabla_{df X} df Y \tag{3.14}$$

$$\mathrm{Exp}_{f(x)} \circ df_x = f \circ \mathrm{Exp}_x\,. \tag{3.15}$$

We now come to the definition of distance and its link with geodesics.

3.3 Distance and Geodesics

3.3.1 Injectivity and parametrization

Let (M, g) be a Riemannian manifold. As the Levi-Civita is uniquely defined, we call geodesic of (M, g) any geodesic of its Levi-Civita connection. Similarly, we define:

Definition 3.7 (Exp and Log maps)**.** We call *Riemannian exponential* the exponential map of the Levi-Civita connection, and define its *injectivity radius* $\mathrm{inj}_M(x)$ at x as the greatest $\epsilon > 0$ such that Exp is a diffeomorphism on the open ball of radius ϵ of T_xM.

The following properties of the exponential map can be proved (Lafontaine, Gallot, et al. 2004, Proposition 2.88).

Proposition 3.5. *Let (M, g) be a Riemannian manifold and $x \in M$.*

- *The differential of Exp_x at 0 is the identity.*
- *$(x, v) \mapsto (x, \mathrm{Exp}_x(v))$ is a smooth diffeomorphism from an open neighborhood of the null section (i.e. "$M \times \{0\}$") in TM to an open neighborhood of the diagonal of $M \times M$.*

The open ball of radius $\mathrm{inj}_M(x)$ on which the Exp map is a diffeomorphism can be maximally extended into a larger star-shaped open set called the *injectivity domain* $\mathrm{Inj}(x) \subseteq T_xM$. The *Riemannian logarithm* is defined as the inverse of the Exp map on this *injectivity domain*. The injectivity radius of M, inj_M, is then the smallest of all injectivity radii at x for $x \in M$. It may not be finite, as in e.g. the hyperbolic space.

By equation (3.9), the squared velocity $E(t) = \frac{1}{2}\|\dot\gamma(t)\|^2$ of any geodesic γ is constant in time:

$$\frac{d}{dt}\langle \dot\gamma(t), \dot\gamma(t)\rangle = 2\langle \nabla_{\dot\gamma(t)}\dot\gamma(t), \dot\gamma(t)\rangle = 0$$

We say that a curve γ is parametrized with constant speed if the function $t \mapsto \|\dot\gamma(t)\|_{\gamma(t)}$ is constant. The above equation thus shows that geodesics are parametrized with constant speed.

Furthermore, if $N \subset M$ is an embedded manifold, (3.13) implies that the geodesics of N are the curves in M whose acceleration is normal to N.

Example 3.6: Geodesics of the hypersphere

For any $x \in S^d \subset \mathbb{R}^{d+1}$, and $v \in T_x S^d$, define the curve on S^d that parametrizes the great circle

$$\gamma : t \mapsto \cos(t\|v\|)x + \sin(t\|v\|)\frac{v}{\|v\|}. \tag{3.16}$$

Then, it is clear that the acceleration $\ddot{\gamma}(t) = -\|v\|^2\gamma(t)$ is normal to S^d. By uniqueness this curve is the geodesic from x with initial velocity v.

The Exp map at x, $v \mapsto \cos(t\|v\|)x + \sin(t\|v\|)\frac{v}{\|v\|}$ is thus well defined and is a diffeomorphism from the open ball of radius π to $S^d \setminus \{-x\}$, i.e. onto the entire sphere except the antipodal point of x. Its inverse is defined for any $y \notin \{x, -x\} \in S^d$ by

$$\mathrm{Log}_x(y) = \arccos(\langle y, x\rangle)\frac{y - \langle y, x\rangle x}{\|y - \langle y, x\rangle x\|}. \tag{3.17}$$

We implement the Riemannian Exp and Log maps in the `HypersphereMetric`.

Implementation trick 3.1. *To improve numerical stability around* 0*, we use a Taylor approximation of the sinc and cosine functions for inputs smaller than* 10^{-6}*. As both functions are even, the squared norm of* v *is computed and taking the square-root is not necessary.*

For a C^1-curve $\gamma : [a, b] \to M$, we now define its *length* L and energy E (also called *action integral* in physics) by:

$$L(\gamma) = \int_a^b \|\dot{\gamma}(t)\| dt, \qquad E(\gamma) = \int_a^b \|\dot{\gamma}(t)\|^2 dt$$

Note that the length does not depend on the parametrization of the curve, while the energy does. Indeed, moving along a path from a to b does not require the same energy if the speed is increased, but the length remains the same. This finally leads to the definition of distance.

3.3.2 Distance and completeness

Definition 3.8 (Riemannian distance)**.** Let (M, g) be a Riemannian manifold, and $x, y \in M$. The *Riemannian distance* between x and y is the lower bound of the lengths of all piecewise smooth curves joining x to y:

$$d(x, y) = \inf\{L(\gamma) \mid \gamma : I \to M \text{ piecewise } C^1,\ \gamma(0) = x,\ \gamma(1) = y\}.$$

We say that γ is *minimizing* if $L(\gamma) = d(x, y)$.

If M is connected*, this distance function is indeed a distance, and the induced topology coincides with that of M (Paulin 2014, Proposition 3.13). We say that a curve $\gamma : [a, b] \to M$ is parametrized by (or resp. proportional to) arc-length if $L(\gamma) = b - a$ (resp. $\exists \lambda > 0, L(\gamma) = \lambda(b - a)$). We now see that the geodesics of (M, g) are the minimizing curves, and further minimize the total energy (Paulin 2014, Proposition 3.14).

Theorem 3.3. *Let* (M, g) *be a Riemannian manifold, and* $\gamma : [a, b] \to M$ *a* C^1 *curve. The following assertions are equivalent*

- γ *is a geodesic;*
- γ *is parametrized with constant velocity and locally minimizing;*

*Defined in Appendix A.

- *γ is locally energy minimizing.*

In particular, as for any $(x, v) \in TM$, the curve $\gamma : t \in \mathrm{Inj}(x) \mapsto \mathrm{Exp}_x(tv)$ is a geodesic, it is locally length-minimizing, so that for any $y \in \mathrm{Exp}_x(\mathrm{Inj}(x))$, $d(x, y) = L(\gamma) = \|v\|$. By definition, $v = \mathrm{Log}_x(y)$, so that

$$d(x, y) = \| \mathrm{Log}_x(y)\|.$$

We add this as a method to the `RiemannianMetric` class of geomstats.

```
def dist(self, point_a, point_b):
        """Riemannian distance between two points"""
        log = self.log(point_b, base_point=point_a)
        return self.norm(log, base_point=point_a)
```

Furthermore, the last point of Theorem 3.3 is a variational principle, i.e. the minimization of a function on the space of paths.

Recall from Definition 3.6 that a Riemannian manifold is called geodesically complete if the Exp map is defined on the whole of TM, meaning that geodesics are defined on all $\mathbb{R}$. On the other hand, the more usual notion of completeness of a metric space is defined as follows:

Definition 3.9 (Complete metric space)**.** A metric space is called complete if every Cauchy sequence converges in that space.

Example 3.7: Completeness of the half-plane

Let the half-plane $P = \{(x, y) \in \mathbb{R}^2 \mid y > 0\}$ be equipped with the canonical Euclidean metric of $\mathbb{R}^2$. Then P is obviously not geodesically complete, as geodesics are straight lines exiting P.

Now define the metric $g(x, y) = \frac{dx^2+dy^2}{y^2}$. (P, g) is now a geodesically complete Riemannian manifold.

The following theorem gives a sufficient condition for Riemannian manifolds to share the same nice properties as complete metric spaces.

Theorem 3.4 (Hopf-Rinow)**.** *Let (M, g) be a Riemannian manifold. If M is geodesically complete, then any two points of M can be joined by a minimizing geodesic.*

Corollary 3.1. *Let (M, g) be a connected Riemannian manifold, and d the Riemannian distance. The following are equivalent*

- *(M, g) is geodesically complete;*
- *every closed and bounded subset of (M, d) is compact;*
- *the metric space (M, d) is complete (as a metric space).*

Remark 3.3 (Infinite dimension)**.** *The Hopf-Rinow theorem is not true in infinite dimension. Counter-examples have been constructed and exhibit manifolds that are complete as metric spaces but for which points exist that cannot be joined by a minimizing geodesic, or even by any geodesic at all. For more details, see Atkin 1975.*

In the previous example of the hypersphere, geodesics are great circles and are known in closed-form. To finish this subsection, we detail our implementation of the Riemannian distance when no closed form solution is available for the geodesics.

3.3.3 Numerical approximations in geomstats

In the most general case, recall that geodesics are defined by an ODE expressed either in a chart or in the ambient space (e.g. Equation (3.7)). Discrete integration methods can thus be used to approximate the exponential map, and optimization algorithms to compute the logarithm. Inspired by Kühnel, Sommer, and Arnaudon 2019, we implemented those methods in geomstats.

Consider the geodesic equation as a coupled system of first-order ODEs:

$$\begin{cases} v(t) &= \dot{\gamma}(t) \\ \dot{v}(t) &= f(v(t), \gamma(t), t) \end{cases} \tag{3.18}$$

where f is a smooth function. A method of the class `Connection` is used to compute the right-hand-side of (3.18), called `geodesic_equation` where the `state` variable is $(\gamma(t), \dot{\gamma}(t))$:

```
def geodesic_equation(self, state, time):
    """Return the right-hide-side of the geodesic equation."""
    position, velocity = state
    gamma = self.christoffels(position)
    equation = -gs.einsum('...kij,...i,...j->...k', gamma, velocity, velocity)
    return gs.stack([velocity, equation])
```

Given initial conditions, a first-order forward Euler scheme, or higher-order Runge-Kutta method can be used to integrate this system.

```
STEP_FUNCTIONS = {'euler': 'euler_step',
                  'rk2': 'rk2_step'}

def euler_step(force, state, time, dt):
    """Compute one step of the Euler approximation."""
    derivatives = force(state, time)
    new_state = state + derivatives * dt
    return new_state

def rk2_step(force, state, time, dt):
    """Compute one step of the rk2 approximation."""
    k1 = force(state, time)
    k2 = force(state + dt / 2 * k1, time + dt / 2)
    new_state = state + dt * k2
    return new_state

def integrate(
        function, initial_state, end_time=1.0, n_steps=10, step='euler'):
    """Compute the flow of a vector field."""

    dt = end_time / n_steps
    states = [initial_state]
    step_function = globals()[STEP_FUNCTIONS[step]]

    current_state = initial_state
    for i in range(n_steps):
        current_state = step_function(
            state=current_state, force=function, time=i * dt, dt=dt)
        states.append(current_state)
    return states
```

We can thus add the following method to the class `Connection` of geomstats to compute the exponential map

```python
    def exp(self, tangent_vec, base_point, n_steps=10, step='euler', **kwargs):
        """Exponential map associated to the affine connection."""
        initial_state = gs.stack([base_point, tangent_vec])
        flow = integrate(
            self.geodesic_equation, initial_state, n_steps=n_steps, step=step)
        exp, velocity = flow[-1]
        return exp
```

On the other hand, the Log map corresponds to a boundary value problem (BVP), and an optimization procedure is required. We choose the *geodesic shooting* method, that solves the following problem ($\mathcal{P}$) which corresponds to an energy minimization. Working in a convex neighborhood, it admits a unique minimizer v^*:

$$\min d^2(\mathrm{Exp}_x(v), y) \tag{$\mathcal{P}$}$$

As in a coordinate chart or in the embedding space, all norms are equivalent, we use the appropriate Euclidean metric (written $\|.\|_2$) for practical purpose.

$$\| \mathrm{Exp}_x(v) - y\|_2^2. \tag{$\mathcal{P}'$}$$

and the problem is solved by gradient descent (GD) until a convergence tolerance ϵ is reached. We use the `minimize` function from the `scipy` function, and use automatic differentiation to compute the gradient of the exponential map.

```python
def log(self, point, base_point, n_steps=N_STEPS, step='euler',
        max_iter=25, verbose=False, tol=gs.atol):
    """Compute logarithm map associated to the affine connection."""
    max_shape = point.shape if point.ndim > base_point.ndim else \
        base_point.shape

    def objective(velocity):
        """Define the objective function."""
        velocity = gs.array(velocity, dtype=base_point.dtype)
        velocity = gs.reshape(velocity, max_shape)
        delta = self.exp(velocity, base_point, n_steps, step) - point
        return gs.sum(delta ** 2)

    objective_with_grad = gs.autograd.value_and_grad(objective)
    tangent_vec = gs.flatten(gs.random.rand(*max_shape))
    res = minimize(
        objective_with_grad, tangent_vec, method='L-BFGS-B', jac=True,
        options={'disp': verbose, 'maxiter': max_iter}, tol=tol)

    tangent_vec = gs.array(res.x, dtype=base_point.dtype)
    tangent_vec = gs.reshape(tangent_vec, max_shape)
    return tangent_vec
```

3.4 Curvature

3.4.1 Definition and Properties

In a Euclidean space, a constant field is parallel along any curve. In a Riemannian manifold in general, parallel transport depends on the curve followed, and there may not exist fields that are parallel along all curves, not even localy. One can investigate the effect of parallel transport along small closed curves. Consider a point $x \in M$ and the closed curves whose tangent velocities at that point span a subspace of dimension two. They introduce a deviation of parallel transport from the identity map of the tangent space at that point, and this deviation can be shown to depend only on a basis of this plane. This is due to the commuting properties of the covariant derivative, i.e. the difference between evaluating ∇_X after ∇_Y and vice-versa. This difference is not sufficient however to define a tensor. The following lemma gives a sufficient condition for a map to define a tensor on a manifold (Lafontaine, Gallot, et al. 2004, Proposition 1.114).

Lemma 3.1 (Tensoriality)**.** *Let $p \in \mathbb{N}$ and $A : \Gamma(TM)^p \to \Gamma(TM)$ be a $C^\infty(M)$-multilinear map, i.e.* $\forall f_1, \dots, f_p \in C^\infty(M), \forall X_1, \dots X_p \in \Gamma(TM), \quad A(f_1 X_1, \dots f_p X_p) = f_1 \dots f_p A(X_1, \dots, X_p)$. *Then for any $x \in M$, the value of $A(X_1, \dots, X_p)$ at x only depends on the value of the X_i's at x.*

Simple computations using the Leibniz rule from the definition of a connection (2) show that the following defines a C^∞-multilinear map (see Lee 2018, Proposition 7 .3 for the computations).

Definition 3.10 (Curvature tensor)**.** Let (M, ∇) be a manifold equipped with a connection. The *curvature* tensor of ∇ is defined as the map from $\Gamma(TM)^3$ to $\Gamma(TM)$ by

$$R(X,Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z \tag{3.19}$$

The curvature tensor of a Riemannian manifold (M, g) is the curvature of its Levi-Civita connection.

In the above definition, the arguments X, Y, Z are vector fields, but the tensoriality Lemma 3.1 allows to write at any $x \in M$, R_x as a map defined on tangent vectors $u, v, w \in T_x M$.

Example 3.8: Curvature of the Euclidean space

In a Euclidean space $\mathbb{R}^d$ with the canonical inner-product, the connection coincides with the directional derivative, which commutes. Therefore $R = 0$.

The following properties help cope with the complexity of this tensor. Proofs can be found in Lafontaine, Gallot, et al. 2004, Proposition 3.5, Gallier and Quaintance 2020, Proposition 16.1, 16 .3.

Proposition 3.6. *Let (M, g) be a Riemannian manifold and R its curvature tensor. The following properties hold*

1. *(Skew-symmetry) The map $(X, Y) \in \Gamma(TM)^2 \mapsto R(X, Y)$ is skew-symmetric.*
2. *(Bianchi's identity) For any $X, Y, Z \in \Gamma(TM)$, we have*
$$R(X,Y)Z + R(Y,Z)X + R(Z,X)Y = 0.$$
3. *(Bianchi's second identity) For any $X, Y, Z \in \Gamma(TM)$, we have*
$$\nabla_X R(Y,Z) + \nabla_Y R(Z,X) + \nabla_Z R(X,Y) = 0.$$
4. *For any isometry $f \in \mathrm{Isom}(M)$ and $X, Y, Z \in \Gamma(TM)$, $R(dfX, dfY)(dfZ) = df\big(R(X,Y)Z\big)$.*
5. *For all $X, Y \in \Gamma(TM)$, the field of linear maps $R(X, Y)$ is skew-symmetric with respect to the metric g, i.e. $\forall W, Z \in \Gamma(TM)$*
$$g(R(X,Y)Z, W) = -g(Z, R(X,Y)W)$$
6. *For all $X, Y, Z, W \in \Gamma(TM)$, we have*
$$g(R(X,Y)Z, W) = g(R(Z,W)X, Y) = g(R(W,Z)Y, X)$$

Note that the compatibility of the metric (equations (3.9), (3.8)) are required for the last two assertions.

3.4.2 Sectional curvature

The previous proposition allows to define the *sectional* curvature for any $x \in M$ and $u, v \in T_xM$ such that u, v are not collinear

$$\kappa_x(u,v) = \frac{\langle R_x(u,v)v, u\rangle}{\|u\|^2\|v\|^2 - \langle u, v\rangle^2} \tag{3.20}$$

From the above properties, the value of κ_x in fact only depends on the plane spanned by (u, v), i.e. for any non-vanishing linear combination $\alpha u + \beta v$, $\kappa(\alpha u + \beta v, v) = \kappa(u, v)$. In fact, this scalar quantity is enough to characterize the curvature of M. Indeed, one can prove from algebraic computations only (see e.g. Kobayashi and Nomizu 1996a, Chapter V, Proposition 1.2) the following

Lemma 3.2. *Let two quadrilinear mappings A, B defined on a vector space V that both verify the properties $\forall u, v, w, z \in V$*

$$A(u,v,w,z) = -A(v,u,w,z) = -A(v,u,z,w)$$
$$A(u,v,w,z) + A(u,w,z,v) + A(u,z,v,w,) = 0$$

and coincide for any two variables $u, v \in V$ in the sense: $A(u,v,u,v) = B(u,v,u,v)$. Then $A = B$.

This applies to the map $(u, v, w, z) \mapsto g(R(w, z)v, u)$ and one can thus show the following theorem (Lafontaine, Gallot, et al. 2004, Theorem 3.8).

Theorem 3.5. *The sectional curvature determines the curvature tensor.*

In particular, if for every $x \in M$, κ has a constant value κ_x on every planes of T_xM, then we can compute the curvature up to this constant. Indeed from the definition (3.20) we have for any $u, v \in T_xM$

$$\langle R(u,v)v, u\rangle = \kappa_x\big(\|u\|^2\|v\|^2 - \langle u, v\rangle^2\big)$$

Consider the maps $A : (u, v, w, z) \mapsto \langle R(u, v)w, z\rangle$ and B defined by

$$B(u,v,w,z) = \big(\langle u, w\rangle\langle v, z\rangle - \langle u, z\rangle\langle v, w\rangle\big)$$

It is clear that the hypotheses of Lemma 3.2 are verified for A and $\kappa_x B$, so we conclude

$$\langle R(u,v)w, z\rangle = (\langle u, w\rangle\langle v, z\rangle - \langle u, z\rangle\langle v, w\rangle)\,\kappa_x. \tag{3.21}$$

For example, eq. (3.21) is always valid in dimension $d = 2$, as there is only one plane in each tangent space.

We now state the surprising result of F. Schur that gives a sufficient condition for formula (3.21) to hold. A proof can be found in Kobayashi and Nomizu 1996a, Theorem 2.2.

Theorem 3.6 (Schur)**.** *Let (M, g) be a connected Riemannian manifold of dimension d. If $d \geq 3$ and if the sectional curvature κ does not depend on the plane but only on the point $x \in M$, then κ is constant (i.e. does not even depend on x).*

Example 3.9: Sectional curvature of the hypersphere and hyperboloid

Recall that the rotation group $SO(d + 1)$ acts on the hypersphere S^d, and suppose that $d \geq 2$. The stabilizer of this action on any $x \in S^d$ is isomorphic to $SO(d)$, whose action on T_xS^d is transitive on 2-planes, i.e. any 2-plane can be mapped by a rotation to any other 2-plane. By assertion 4. of Proposition 3.6,

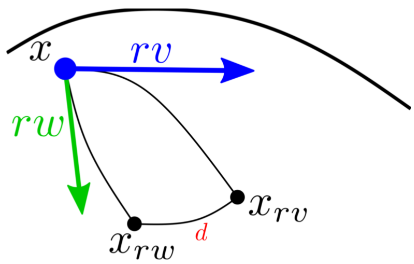


Figure 11: Drawing of the distortion of geodesics compared to Euclidean tangent vectors. Arrows represent tangent vectors, thin black lines represent geodesics and we use the notation $x_v = \mathrm{Exp}_x(v)$. Curvature modifies the geodesic distance d compared with the distance between rv and rw in T_xM (adapted from Paulin 2014, Section 3.6.3)

.

the sectional curvature is preserved from one plane to another by rotations, it is therefore constant on the whole T_xM, write this value κ_x. If $d \geq 3$, we can use Theorem 3.6 to conclude it is constant on S^d. Of course, it is also constant for $d = 2$, as we can use an isometry to map the curvature tensor from one point to any other.

The same argument, using this time the group $O(1, d)$ of isometries of the Lorentz product, that acts transitively on the hyperboloid H_+^d, shows that this space also has constant curvature.

Definition 3.11 (Constant curvature)**.** Let (M, g) be a Riemannian manifold. (M, g) is said to have *constant* (resp. *negative*, resp. *positive*) curvature if it has constant (resp. negative, resp. positive) sectional curvature.

In fact, the (complete simply connected) constant curvature spaces are all isometric to one of the Euclidean space (flat), the hypersphere (positive curvature) or the hyperbolic space (negative curvature) (see e.g . Lafontaine, Gallot, et al. 2004, Theorem 3.82). These three examples thus describe the entire class of constant curvature spaces (up to covering and isometry).

To finish this section, we may now explain how curvature modifies the correspondence between distances in tangent spaces and Riemannian distances on a manifold. Indeed, let (M, g) be a Riemannian manifolds and consider two orthogonal tangent vectors v, w at some $x \in M$. In the vector space T_xM, the distance (induced by the metric at x) between rv and rw for some $r > 0$ is of course $r\sqrt{2}$. Now map these vectors to the manifold using the exponential map, then the Riemannian distance between $\mathrm{Exp}_x(rv)$ and $\mathrm{Exp}_x(rw)$ may increase or decrease compared to the distance in the tangent space. Curvature is the fundamental tool that allows to quantify these variations, and the sign of the sectional curvature tells whether geodesics accumulate or grow apart (Figure 11).

Theorem 3.7. *Let (M, g) be a Riemannian manifold, R its curvature tensor, κ its sectional curvature, $x \in M$ and γ_v, γ_w two geodesics starting from x with initial velocities $v, w \in T_xM$. Then for $r \to 0$,*

$$d(\gamma_v(r), \gamma_w(r))^2 = r^2\|v - w\|^2 - \frac{r^4}{3}\langle R(v, w)w, v\rangle + O(r^5). \tag{3.22}$$

Moreover, if v, w are orthonormal

$$d(\gamma_v(r), \gamma_w(r)) = \sqrt{2}r\left(1 - \frac{\kappa(v, w)}{12}r^2\right) + O(r^4). \tag{3.23}$$

We now understand the importance of the sign of the sectional curvature: if $\kappa > 0$, geodesics get closer to one another, while if $\kappa < 0$, they grow apart. Using this theorem and formulas for geodesics on the hypersphere in Example 3.6, we can compute a Taylor expansion of $d(\gamma_v, \gamma_w)$ and identify the coefficients, to find that $\kappa = 1$. Similarly, for the hyperbolic space $\kappa = -1$.

Finally, we state two important theorems that give a hint about how the sign of the curvature determines the geometry. These theorems also show how curvature and topology may be intertwined. See Lafontaine, Gallot, et al. 2004, Theorem 3.87 for a proof.

Theorem 3.8 (Cartan-Hadamard)**.** *Let (M, g) be a complete connected Riemannian manifold with non-positive sectional curvature. Then the exponential map is a Riemannian covering, i.e. a covering that is a local isometry. This means that if M is simply connected, then M is diffeomorphic to $\mathbb{R}^d$, and any two points are joined by a unique minimizing geodesic.*

Remark 3.4 (Infinite dimension)**.** *This theorem carries over to infinite dimensional Riemannian Hilbert manifolds, i.e. whose atlases are valued in a Hilbert space (McAlpin 1965).*

On the contrary (Berger 2003, Theorem 63):

Theorem 3.9 (Particular case of Bonnet-Myers theorem)**.** *Let (M, g) be a complete connected Riemannian manifold with sectional curvature bounded below $\kappa > \frac{1}{r^2}$ for some $r > 0$. Then the diameter of M (i.e. the largest distance between points in M) is bounded above by πr and M is compact.*

This ends our exposition of Riemannian metrics. In the next section, we focus on a particular class of manifolds that play a fundamental role in geometry and in many applications.

4 Lie groups

We first define the notions of Lie groups, subgroups and algebras, then the exponential map of Lie groups, and finally introduce smooth actions of Lie groups and homogeneous spaces. The metrics that can be defined on Lie groups, and their implementation in geomstats are be covered in Subsection 4.3. We restrict to finite-dimensional Lie groups and only give a few remarks about infinite dimension.

4.1 Lie groups, Lie algebras and Lie subgroups

Definition 4.1 (Lie group)**.** A Lie group is a group* $(G, \cdot)$ such that G is also a finite dimensional smooth manifold, and the group and differential structures are compatible, in the sense that the group law $\cdot$ and the inverse map $g \mapsto g^{-1}$ are smooth.

We generally omit $\cdot$ in the notation and simply write the group composition as a multiplication. Let e denote the neutral element, or identity of G.

Remark 4.1 (Infinite dimension)**.** *Infinite dimensional groups with a smooth structure appear naturally, for example as the set of diffeomorphisms of a smooth manifolds. However, although some properties of finite dimensional Lie groups generalize to infinite dimension, many don't. See Milnor 1984 for a good treatment of the topic. Unless otherwise specified, we restrict our exposition to finite dimension.*

*Defined in Appendix A.

For any $g \in G$, we define the *left* and *right translation* maps L_g and R_g by

$$L_g : h \in G \mapsto gh \in G \qquad R_g : h \in G \mapsto hg^{-1}.$$

By the definition of a Lie group, L_g and R_g are diffeomorphisms of G and their differential are linear isomorphisms of tangent spaces. This means that all the tangent spaces of G are identical to T_eG, and there is a canonical way of mapping vectors from T_gG to T_eG, by $L_{g^{-1}}$ for any $g \in G$.

This fact, that could be rephrased as "the tangent bundle of a Lie group is trivial", means that TG is diffeomorphic to the product $G \times T_eG$, and is of fundamental importance to characterize the geometry of a Lie group, and is very handy for the implementation. Furthermore, the maps $g \mapsto L_g$ and $g \mapsto R_g$ are group homomorphisms* between $(G, \cdot)$ and $(\mathrm{Diff}(G), \circ)$, the group of diffeomorphisms of G with the composition of maps as group law.

Definition 4.2 (Invariant vector field)**.** A vector field $X \in \Gamma(TG)$ is *left-invariant* if $\forall g, h \in G$,

$$X_{gh} = dL_g X_h$$

Let $\mathcal{L}(G)$ denote the set of all left-invariant vector fields.

Example 4.1: General Linear group

The real general linear group $GL(n)$ plays a fundamental role as all the groups implemented in geomstats and encountered in the applications are subgroups of $GL(n)$. It is defined as the set of invertible matrices of size n

$$GL(n) = \{A \in M_n(\mathbb{R}) \mid \det(A) \neq 0\},$$

and as an open set of the vector space $M_n(\mathbb{R}) \simeq \mathbb{R}^{n^2}$ it is a smooth manifold.

The group law is the matrix multiplication, that can be written as a polynomial of the matrices coefficients and is thus smooth. Thanks to Cramer's formula $A^{-1} = \det(A)^{-1} Co(A)^T$, where Co denotes the matrix formed by all cofactors, the inversion map is also smooth. Therefore, $GL(n)$ is a Lie group.

Definition 4.3 (Lie algebra)**.** A real *Lie algebra* is a real vector space $\mathfrak{g}$ equipped with a bilinear map $[\cdot, \cdot]$ that verifies

- (Skew-symmetry) $\forall x, y \in \mathfrak{g}, \quad [x, y] = -[y, x]$,
- (Jacobi identity) $\forall x, y \in \mathfrak{g}, \quad [x, [y, z]] + [y, [z, x]] + [z, [y, x]] = 0$.

Example 4.2: Vector fields

The space of vector fields $\Gamma(TM)$ of a manifold M equipped with the Lie bracket of vector field defined in eq. (2.2) page14 is an infinite dimensional Lie algebra.

Example 4.3: Matrix algebra

The algebra of squared matrices $M_n(\mathbb{R})$ equipped with the commutator

$$\forall A, B \in M_n(\mathbb{R}), \quad [A, B] = AB - BA$$

is a Lie algebra.

Implementation in geomstats As the Lie groups we are working with are all matrix groups, we implement an abstract class `MatrixLieGroup` to gather the properties of a Lie group: its identity, composition law, translation map, exponential and logarithm maps. Morever, it is useful to have the Lie algebra implemented as a separate class and set as an attribute of the Lie group. This class is a subclass of `VectorSpace` and implements a `belongs` and `projection` method. This allows for example to write the methods `is_tangent` and `to_tangent` as follows:

```python
def tangent_translation(self, point, left_or_right='left', inverse=False):
    """Return the differential map of the right or left translation map."""
    point_ = self.inverse(point) if inverse else point
    if left_or_right == 'left':
        return lambda tan_vec: self.compose(point_, tan_vec)
    return lambda tan_vec: self.compose(tan_vec, point_)

def is_tangent(self, vector, base_point=None, atol=gs.atol):
    """Check whether a vector is tangent at base point."""
    if base_point is None:
        base_point = self.identity

    if gs.allclose(base_point, self.identity):
        tangent_vec_at_id = vector
    else:
        tangent_vec_at_id = self.tangent_translation(
            base_point, inverse=True)(vector)
    return self.lie_algebra.belongs(tangent_vec_at_id, atol)

def to_tangent(self, vector, base_point=None):
    """Project a vector onto the tangent space at a base point."""
    if base_point is None:
        return self.lie_algebra.projection(vector)
    tangent_vec_at_id = self.tangent_translation(
        base_point, inverse=True)(vector)
    projected = self.lie_algebra.projection(tangent_vec_at_id)
    return self.tangent_translation(base_point)(projected)
```

Example 4.4: Implementation of the general linear group

The general linear group (Example 4.1) is the archetype of matrix Lie group, and thus created as a subclass of `MatrixLieGroup`. It is defined as an open set of the matrix space, so it also inherits from `OpenSet`.

```python
class GeneralLinear(MatrixLieGroup, OpenSet):
    """Class for the general linear group GL(n)."""

    def __init__(self, n, **kwargs):
        if 'dim' not in kwargs.keys():
            kwargs['dim'] = n ** 2
        super().__init__(
            ambient_space=Matrices(n, n), n=n, **kwargs)

    def belongs(self, point, atol=gs.atol):
        """Check if a matrix is invertible and of size n."""
        has_right_size = self.ambient_space.belongs(point)
        if gs.all(has_right_size):
            det = gs.linalg.det(point)
            return gs.abs(det) > atol
        return has_right_size

    def projection(self, point):
        """Project a matrix to the general linear group."""
        belongs = self.belongs(point)
        regularization = gs.einsum(
            '...,ij->...ij', gs.where(~belongs, gs.atol, 0.), self.identity)
        projected = point + regularization
        return projected
```

The set of left-invariant vector fields is fundamental when studying Lie groups it is closed under Lie brackets and boils down to the tangent space at the identity, as stated by the following theorem (Lee 2003, Proposition 8.33 and Theorem 8.37).

Theorem 4.1 (Left-invariant vector fields)**.** *Let G be a Lie group of finite dimension d.*

(1) The space of left-invariant vector fields $\mathcal{L}(G)$ of a Lie group G is a sub-algebra of $\Gamma(TG)$. This means that the bracket of two left-invariant vector fields is also left-invariant.

(2) As a vector space, $\mathcal{L}(G)$ is isomorphic to T_eG, the tangent space at the identity of G. This implies that $\mathcal{L}(G)$ is finite dimensional.

Indeed, the map $X \in \mathcal{L}(G) \mapsto X_e \in T_eG$ is linear and has inverse $x \mapsto \tilde{x}$ where we define $\tilde{x}$ to be the left-invariant vector field

$$\tilde{x} : g \in G \longmapsto dL_g x \in T_gG.$$

We can thus define the bracket on T_eG: $[x, y] \triangleq [\tilde{x}, \tilde{y}]_e$, which turns T_eG into a Lie algebra that is isomorphic (as Lie algebras) to $\mathcal{L}(G)$. We call T_eG *the* Lie algebra of G and denote it $\mathfrak{g} = T_eG$.

Thankfully, a linear representation of G allows to compute the Lie bracket without requiring the handling of vector fields. Indeed, define the conjugation map of G, for any $g \in G$ by $C_g : h \mapsto ghg^{-1}$, and its differential at the identity, called adjoint representation of G:

$$\mathrm{AD}_g : \mathfrak{g} \mapsto \mathfrak{g}.$$

Then the differential of $g \mapsto \mathrm{AD}_g$, written ad is called the adjoint representation of $\mathfrak{g}$, and one can show that it coincides with the Lie bracket of $\mathfrak{g}$:

$$\mathrm{ad}_x(y) = [x, y].$$

Example 4.5: General linear Lie algebra

Recall from Example 4.1 that $GL(n)$ is a Lie group, whose differentiable structure comes from its embedding in $M_n(\mathbb{R})$. Thus, by (1) of Theorem 2.2 (page 12), its tangent space at the identity is the entire matrix space, and as the Lie algebra of $GL(n)$ it is written $\mathfrak{gl}(n)$. We now compute explicitely AD and ad and verify that the bracket of $\mathfrak{gl}(n)$ coincides with the commutator defined in Example 4.3.

The conjugation map is the restriction to $GL(n)$ of the linear map $h \mapsto ghg^{-1}$ of $M_n(\mathbb{R})$ for any $g \in GL(n)$. Its differential is thus for any $X \in M_n(\mathbb{R})$

$$\mathrm{AD}_g(X) = gXg^{-1}.$$

Now consider a curve $c : (-\epsilon, \epsilon) \to GL(n)$ such that $c(0) = I_n$ and $c'(0) = X \in M_n(\mathbb{R})$. For any $Y \in M_n(\mathbb{R})$ we have

$$\mathrm{ad}_X Y = d(g \mapsto \mathrm{AD}_g(Y))_{I_n} = \left.\frac{d}{dt}\right|_{t=0} (c(t)Yc(t)^{-1}) = XY - YX$$

To summarize,

- The Lie algebra of a Lie group is its tangent space at identity T_eG or equivalently, the algebra of left-invariant vector fields.
- The Lie bracket on T_eG is defined via the adjoint representation and coincides with that of left-invariant vector fields.
- In the case of linear Lie algebras (i.e. subalgebras of $M_n(\mathbb{R})$), the Lie bracket coincides with the matrix commutator.

Remark 4.2. *Note that right-invariant vector fields can be defined analogously as left-invariant vector fields, and their set forms a Lie algebra. However, the Lie bracket of right-invariant vector fields coincides with the opposite of the adjoint representation. For practical and historical reasons, right-invariant fields are used in infinite dimension while left-invariant fields are used in finite dimension.*

As all the next examples will be subgroups of the general linear group, we state a result by von Neumann and Cartan known as the *closed-subgroup theorem* that gives a necessary and sufficient condition for a subgroup of a Lie group G to be an embedded Lie subgroup, i.e. a Lie group with differential structure agreeing with that of G, and such that the inclusion map is smooth (Cartan 1930, Section III, paragraph 26). See Lee 2003, Theorem 20.12 for a proof.

Theorem 4.2 (Cartan-von-Neumann)**.** *Let G be a Lie group. Any closed sugroup $H \subset G$ of G is a Lie subgroup of G. Conversely, any Lie subgroup of G is closed.*

One can also show that the Lie algebra of a subgroup G is a Lie subalgebra of the Lie algebra $\mathfrak{g}$ of G. Thus, they share the same bracket. Therefore, all Lie algebras considered in geomstats use the matrix commutator as bracket.

Example 4.6: Implementation of the special orthogonal group

Recall that the special orthogonal group is the set of orthogonal matrices of positive determinant. It is clear that it is stable by multiplication, hence it is a subgroup of $GL(n)$.

Furthermore, $SO(n) = f^{-1}(I_n) \cap \det^{-1}(1)$, where $f : A \mapsto A^\top A$ and det are continuous maps. $SO(n)$ is thus closed in $M_n(\mathbb{R})$, and is hence a Lie subgroup of $GL(n)$.

Moreover, from Example 2.10 (page 13), its Lie algebra is

$$\mathfrak{so}(n) \triangleq T_{I_n}SO(n) = \{A \in M_n(\mathbb{R}) \mid A^\top + A = 0\} = \mathrm{Skew}(n).$$

In geomstats, $SO(n)$ is implemented as a subclass of the `MatrixLieGroup` class and as it is embedded in $GL_+(n)$ (the subgroup of $GL(n)$ formed by matrices with positive determinent), it also inherits from the `LevelSet` class. This allows to inherit the composition method, but the inverse method is overridden by the transposition.

```
class SpecialOrthogonalMatrices(MatrixLieGroup, LevelSet):
    """Class for special orthogonal group."""

    def __init__(self, n):
        matrices = Matrices(n, n)
        gln = GeneralLinear(n, positive_det=True)
        super().__init__(
            dim=int((n * (n - 1)) / 2), n=n, value=gs.eye(n),
            lie_algebra=SkewSymmetricMatrices(n=n), embedding_space=gln,
            submersion=lambda x: matrices.mul(matrices.transpose(x), x),
            tangent_submersion=lambda v, x: 2 * matrices.to_symmetric(
                matrices.mul(matrices.transpose(x), v)))
        self.bi_invariant_metric = BiInvariantMetric(group=self)
        self.metric = self.bi_invariant_metric

    @classmethod
    def inverse(cls, point):
        """Return the transpose matrix of point."""
        return cls.transpose(point)

    def projection(self, point):
        """Project a matrix on SO(n) by minimizing the Frobenius norm."""
        aux_mat = self.submersion(point)
        inv_sqrt_mat = SymmetricMatrices.powerm(aux_mat, - 1 / 2)
        rotation_mat = Matrices.mul(point, inv_sqrt_mat)
        det = gs.linalg.det(rotation_mat)
        return utils.flip_determinant(rotation_mat, det)
```

And the Lie algebra is implemented as follows.

```
class SkewSymmetricMatrices(MatrixLieAlgebra):
    """Class for skew-symmetric matrices."""

    def __init__(self, n):
        dim = int(n * (n - 1) / 2)
        super().__init__(dim, n)
        self.ambient_space = Matrices(n, n)

    def belongs(self, mat, atol=gs.atol):
        """Evaluate if mat is a skew-symmetric matrix."""
        has_right_shape = self.ambient_space.belongs(mat)
        if has_right_shape:
            return Matrices.equal(mat, - Matrices.transpose(mat), atol=atol)
        return False

    @classmethod
    def projection(cls, mat):
        """Compute the skew-symmetric component of a matrix."""
        return 1 / 2 * (mat - Matrices.transpose(mat))
```

Example 4.7: Implementation of the special Euclidean group

We now introduce a group that is ubiquitous in applications. This group is defined as the set of direct isometries - or rigid-body transformations - of $\mathbb{R}^n$, i.e. the linear transformations of the affine space $\mathbb{R}^n$ that preserve its canonical inner-product. Such transformation ρ can be decomposed in a rotation part and a translation part: $\rho(x) = Rx + u$, where $R \in SO(n)$ and $x, u \in \mathbb{R}^n$. Define

$$SE(n) = \{(R, u) \mid R \in SO(n), u \in \mathbb{R}^n\}.$$

Now, the composition of two isometries $\rho = (R, u), \rho'(R', u')$ remains an isometry:

$$\rho \circ \rho'(x) = RR'x + Ru' + u,$$

and can be written $\rho \circ \rho' = (RR', Ru' + u)$. This suggests the representation of $SE(n)$ in *homogeneous* coordinates

$$\rho = \begin{pmatrix} R & u \\ 0 & 1 \end{pmatrix} \in GL(n+1) \tag{4.1}$$

The composition of isometries then corresponds to matrix multiplication, and $SE(n)$ is now a Lie subgroup of $GL(n+1)$. Its Lie algebra is

$$\mathfrak{se}(n) = \left\{ \begin{pmatrix} S & v \\ 0 & 0 \end{pmatrix} \mid S \in \mathrm{Skew}(n), v \in \mathbb{R}^n \right\}.$$

Thus its dimension is $\frac{n(n+1)}{2}$. In geomstats, $SE(n)$ inherits from `MatrixLieGroup` and is embedded in $GL_+(n+1)$ with overridden `inverse` function. We used a utility function that builds a block matrix according to equation (4.1) The submersion used is the map

$$\begin{pmatrix} R & u \\ v^\top & c \end{pmatrix} \in GL_+(n+1) \mapsto (R^\top R, v, c) \in M_n(\mathbb{R}) \times \mathbb{R}^n \times \mathbb{R},$$

so that $SE(n) = f^{-1}\{(I_n, 0, 1)\}$. It is not printed in the code below in the interest of space.

```python
class SpecialEuclideanMatrices(MatrixLieGroup, LevelSet):
    """Class for special Euclidean group."""

    def __init__(self, n):
        super().__init__(
            n=n + 1, dim=int((n * (n + 1)) / 2),
            embedding_space=GeneralLinear(n + 1, positive_det=True),
            submersion=submersion, value=gs.eye(n + 1),
            tangent_submersion=tangent_submersion,
            lie_algebra=SpecialEuclideanMatrixLieAlgebra(n=n))
        self.rotations = SpecialOrthogonal(n=n)
        self.translations = Euclidean(dim=n)
        self.n = n

        self.left_canonical_metric = \
            SpecialEuclideanMatrixCannonicalLeftMetric(group=self)
        self.metric = self.left_canonical_metric

    @property
    def identity(self):
        """Return the identity matrix."""
        return gs.eye(self.n + 1, self.n + 1)

    def inverse(self, point):
        """Return the inverse of a point."""
        n = self.n
        transposed_rot = self.transpose(point[..., :n, :n])
        translation = point[..., :n, -1]
        translation = gs.einsum(
            '...ij,...j', transposed_rot, translation)
        return homogeneous_representation(
            transposed_rot, -translation, point.shape)
```

Remark 4.3. *Note that as a manifold, $SE(n)$ defined in the above example corresponds to the product manifold $SO(n) \times \mathbb{R}^n$ (Example 2.6 page 10), but as a Lie group, it is the semi-direct product $SO(n) \ltimes \mathbb{R}^n$, because the rotation part acts on the translation part in the composition rule.*

4.2 The exponential map

We now study the flow of left-invariant vector fields of a Lie group. For more details and proofs, we refer the reader to Gallier and Quaintance 2020, Chapter 18. It allows to canonically map the Lie algebra to its Lie group. First, the left-invariance translates into a commutation property of the flow with the left translation map:

Proposition 4.1. *Let G be a Lie group and $X \in \mathcal{L}(G)$. Then X is complete, and if $(\phi_t)_t$ is its flow, then for any $t \in \mathbb{R}$ and $g, g' \in G$ we have*

$$\phi_t(gg') = g\phi_t(g'), \quad i.e. \quad \phi_t \circ L_g = L_g \circ \phi_t.$$

This allows the following

Definition 4.4 (Exponential map)**.** We call exponential map, and write $\exp : \mathfrak{g} \to G$ the map defined by $x \mapsto \phi_1(e)$ where $(\phi_t)_t$ is the flow of the left-invariant vector fields $\tilde{x}$.

Remark 4.4. *This is the second definition of exponential map we encounter, where the first was the Riemannian exponential defined by the geodesic flow of a metric. We will refer to the one canonically defined on a Lie group as the group exponential and use a lowercase exp.*

Example 4.8: Exponential map of the general linear group

The fundamental case is of course again that of the general linear group. Recall that its Lie algebra $\mathfrak{gl}(n) = M_n(\mathbb{R})$ is the set of square matrices, and the group law is the (linear) matrix multiplication. Therefore, a left-invariant vector field $\tilde{X}$ associated to $X \in \mathfrak{gl}(n)$ is defined by

$$g \mapsto gX$$

Let γ be the integral curve from the identity I_n. This means that γ is solution to the ODE defined on $\mathbb{R}$ with initial condition $\gamma(0) = I_n$

$$\gamma'(t) = \gamma(t)X$$

It is well known that the unique solution is the matrix exponential, defined by the series

$$e^{tX} = \sum_{k=0}^{\infty} \frac{t^k}{k!} X^k.$$

Thus $\exp(X) = e^X$.

We now state some of the fundamental properties of the exponential map (see e.g. Gallier and Quaintance 2020, Proposition 18.6-18.7-18.13).

Proposition 4.2. *For any Lie group G, the exponential map $\exp : \mathfrak{g} \to G$ is smooth and a local diffeomorphism at 0.*

The inverse of the group exponential, defined locally, is called *logarithm* map and is valued in $\mathfrak{g} = T_eG$. It thus allows to map data defined on G to its Lie algebra $\mathfrak{g}$, which is a vector space! This fact is at the basis of many algorithms that handle Lie group data.

Remark 4.5 (Infinite dimension)**.** *Proposition 4.2 is not true in infinite dimension. For example, if $G = \mathrm{Diff}(M)$, the exp map is not even surjective in any neighborhood of the identity (Schmid 2004).*

Proposition 4.3. *Let G, H be a Lie group, and $f : G \to H$ a Lie group homomorphism. Then*

$$f \circ \exp = \exp \circ df_e \tag{4.2}$$

In particular, if $G = H$ and $f = C_g$, we have

$$\exp(t\,\mathrm{AD}_g(u)) = g \exp(tu) g^{-1} = C_g(\exp(tu)). \tag{4.3}$$

See Lee 2003, Proposition 20.8 for a proof. The commutation property (4.2) can be depicted by saying that the following diagram commutes, meaning both paths leading from the top left space to the bottom right one are equivalent:

$$\begin{tikzcd} \mathfrak{g} \arrow[r, "\exp"] \arrow[d, "df_e"] & G \arrow[d, "f"] \\ \mathfrak{h} \arrow[r, "\exp"] & H \end{tikzcd}$$

Using $f = L_g$, or equivalently (thanks to eq. (4.3)) $f = R_g$, we define the exponential map at any point $g \in G$ so that it fulfills the above properties. For any $v \in T_gG$

$$\exp_g(v) = L_g \circ \exp \circ (dL_g)^{-1} = R_{g^{-1}} \circ \exp \circ dR_g.$$

where differentiation is at e, and so that the group exp is now understood as $\exp_e$. In geomstats we use the implementation of the matrix exponential and logarithm from the backends and include it in the `MatrixLieGroup` class.

Moreover, we can recover the whole connected component of the identity from the Lie algebra:

Theorem 4.3. *If G is a Lie group and G_0 is the connected component of e, then G_0 is generated by* $\exp(\mathfrak{g})$.

In particular (see Gallier and Quaintance 2020, Theorem 1.6 and 1.12), we have the following

Proposition 4.4. *$SO(n)$ and $SE(n)$*

- *The exponential map* $\exp : \mathfrak{so}(n) \to SO(n)$ *is surjective,*
- *The exponential map* $\exp : \mathfrak{se}(n) \to SE(n)$ *is surjective.*

Note that by applying the commutation property (4.2) to the inclusion map, the exponential maps of $SE(n)$ and $SO(n)$ coincide with the restriction of the exponential map of $GL(n)$, i.e. the matrix exponential.

Finally, recall that flows of complete vector fields are one-parameters subgroups of $\mathrm{Diff}(M)$. Define now a one-parameter subgroup of G as a Lie group homomorphism $t \in \mathbb{R} \mapsto \gamma_t \in G$, written $(\gamma_t)_{t\in\mathbb{R}}$. Then by a change of variable in the flow equation, it is clear that $t \mapsto \exp(tX)$ is a one-parameter subgroup for any $X \in \mathfrak{g}$. In fact, all one-parameter subgroups are of this form and the maps:

$$\theta : (\gamma_t)_{t\in\mathbb{R}} \mapsto \frac{d}{dt}\Big|_{t=0} \gamma_t; \qquad \theta' : X \mapsto \big(\exp(tX)\big)_{t\in\mathbb{R}}$$

are inverse to each other, between $\mathfrak{g}$ and the set of one-parameter subgroups.

Before closing this section, we illustrate it with a representation of the one-parameter subgroups of $SE(2)$.

Example 4.9: Curve in $SE(2)$

Recall that $SE(2)$ is the group of rotation-translation transformations, so that a smooth curve of $SE(2)$ can represent the motion of a rigid-body in a plane. An element of $SE(2)$ can be represented by the application of the rotation part to the canonical orthonormal frame of the $2d$-plane, while the translation part is canonically mapped to a point in the $2d$-plane. See appendix B.3 for the code and different initial vectors.

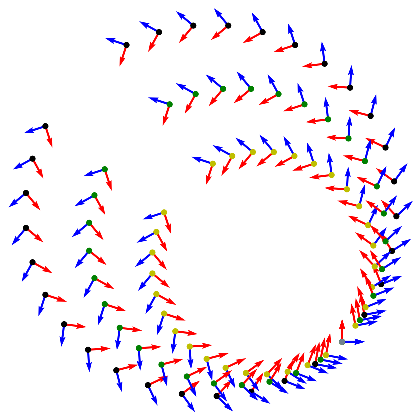


Figure 12: One-parameter subgroup of the special Euclidean group with $n = 2$.

4.3 Invariant metrics on Lie groups

4.3.1 Rationale

Different geometric structures are compatible with the group structure. For instance, the group exponential defined in Section 4 can be defined as the exponential map of a connection on G, implying that geodesics are one-parameter subgroups. This connection is known as the canonical Cartan connection, and is of practical interest as in linear groups geodesics can be computed in closed-form using the matrix exponential.

The canonical Cartan connection may however not be the Levi-Civita connection of any metric, for more details we refer the reader to Xavier Pennec, Sommer, et al. 2020, Chapter 5. In this section we focus on the study of invariant metrics on Lie groups, i.e. metrics for which the left (or right) translation map is an isometry. This case is fundamental in geometric mechanics (Kolev 2004) and has been studied in depth since the foundational papers of Arnold 1966 and Milnor 1976. Using left-invariant vector fields, one can compute explicitly the LC connection, allowing to rewrite the geodesic equation. This fundamental idea, known as Euler-Poincaré reduction, is that the geodesic equation can be expressed entirely in the Lie algebra thanks to the symmetry of left-invariance (Marsden and Ratiu 2009), alleviating the burden of coordinate charts.

We derive here a similar reduction of the parallel transport equation, resulting in a stable and efficient implementation of parallel transport in geomstats. Finally, knowing the connection, one can also compute algebraically the curvature of the space. We exemplify these results with an anisotropic metric on the special Euclidean groups $SE(2)$ and $SE(3)$. Part of the material and results of this section were presented at the GSI 2021 conference in Guigui and Xavier Pennec 2021.

4.3.2 Definitions and computations of the connection

Let G be a Lie group. Recall that the Lie algebra of left-invariant vector fields $\mathcal{L}(G)$ and the tangent space at the identity $\mathfrak{g}$ can be identified, and in this section we will write $\tilde{x}$ the left-invariant field generated by $x \in \mathfrak{g}$: $\forall g \in G,\ \tilde{x}_g = dL_g x$.

Definition 4.5 (Invariant metric)**.** Let G be a Lie group. A Riemannian metric $\langle \cdot, \cdot \rangle$ on G is called *left-invariant* if the differential map of the left translation is an isometry between tangent spaces, that is

$$\forall g, h \in G, \forall u, v \in T_g G, \quad \langle u, v \rangle_g = \langle dL_h u, dL_h v \rangle_{hg}.$$

It is thus uniquely determined by an inner product on the tangent space at the identity $T_e G = \mathfrak{g}$ of G. Similarly, a *right-invariant* metric is such that the differential of the right translation is an isometry. A metric is called *bi-invariant* if it is both left and right-invariant.

Moreover, let $(e_1, \ldots, e_n)$ be an orthonormal basis of $\mathfrak{g}$, and the associated left-invariant vector fields $\tilde{e_i}$. As dL_g is an isomorphism, $(\tilde{e}_{1,g}, \ldots, \tilde{e}_{n,g})$ form a basis of $T_g G$ for any $g \in G$, so for any $X \in \Gamma(G)$ one can write $X_g = f^i(g)\tilde{e}_{i,g}$ where for $i = 1, \ldots, n$ $g \mapsto f^i(g)$ is a smooth real-valued function on G. Any vector field on G can thus be expressed as a linear combination of the $\tilde{e}_i$ with functional coefficients.

Example 4.10: Invarirant metric on the special orthogonal group

Consider the special orthogonal group endowed with the restriction of the Frobenius metric of Example 3.5. Let $P, Q \in SO(n)$ and $A, B \in T_P SO(n) = P\,\mathrm{Skew}(n)$. Then it is clear that

$$\langle dL_Q A, dL_Q B \rangle = \mathrm{tr}((QA)^\top QB) = \mathrm{tr}(A^\top B) = \langle A, B \rangle,$$

and similarly that $\langle dR_Q A, dR_Q B \rangle = \langle A, B \rangle$. Therefore, the Frobenius metric is bi-invariant on $SO(n)$.

Example 4.11: Invariant metric on the special Euclidean group

Now, consider the special Euclidean group endowed with the restriction of the Frobenius metric. As in the previous example, it is easy to show that it is left-invariant. However, it is not right-invariant. Indeed, let $g = (P, t) \in SE(n)$ and $x = (S, u), y = (T, v) \in \mathfrak{se}(n) = \mathrm{Skew}(n) \oplus \mathbb{R}^n$. Then

$$\begin{aligned}\langle dR_g x, dR_g y\rangle &= \langle \begin{pmatrix} SP^\top & u - SP^\top t \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} TP^\top & v - TP^\top t \\ 0 & 0 \end{pmatrix} \rangle \\ &= \mathrm{tr}(S^\top T) + u^T v + t^\top P S^\top T P^\top t \\ &= \langle x, y\rangle + t^\top P S^\top T P^\top t\end{aligned}$$

So that one can find $t \neq 0$ and $S, T \in \mathrm{Skew}(n)$ such that $\langle dR_g x, dR_g y\rangle \neq \langle x, y\rangle$. Therefore the metric is not bi-invariant. In fact one can show that there does not exist any bi-invariant metric on $SE(n)$ for $n \neq 1$, and there exists a bi-invariant pseudo-metric for $n = 3$ (N. Miolane and X. Pennec 2015).

Definition 4.6 (Dual adjoint)**.** Define the metric dual adjoint map on $\mathfrak{g}$ as the unique map that verifies

$$\forall a, b, c \in \mathfrak{g}, \langle \mathrm{ad}_a^*(b), c\rangle = \langle b, \mathrm{ad}_a(c)\rangle = \langle [a, c], b\rangle.$$

As the bracket can be computed explicitly in the Lie algebra, so can ad^* thanks to the orthonormal basis of $\mathfrak{g}$. Now let ∇ be the Levi-Civita connection associated to the metric. It is also left-invariant and can be characterized by a bi-linear form on $\mathfrak{g}$ that verifies $\forall x, y \in \mathfrak{g}$ (Xavier Pennec and Arsigny 2013; Gallier and Quaintance 2020):

$$\alpha(x, y) \triangleq (\nabla_{\tilde{x}} \tilde{y})_e = \frac{1}{2}\big([x, y] - \mathrm{ad}_x^*(y) - \mathrm{ad}_y^*(x)\big) \tag{4.4}$$

Indeed by the left-invariance, for two left-invariant vector fields $X = \tilde{x}, Y = \tilde{y} \in \mathcal{L}(G)$, the map $g \mapsto \langle X, Y\rangle_g$ is constant, so for any vector field $Z = \tilde{z}$ we have $Z(\langle X, Y\rangle) = 0$. Kozsul formula thus becomes

$$\begin{aligned} 2\langle \nabla_X Y, Z\rangle &= \langle [X, Y], Z\rangle - \langle [Y, Z], X\rangle - \langle [X, Z], Y\rangle \qquad (4.5)\\ 2\langle \nabla_X Y, Z\rangle_e &= \langle [x, y], z\rangle_e - \langle \mathrm{ad}_y(z), x\rangle_e - \langle \mathrm{ad}_x(z), y\rangle_e \\ 2\langle \alpha(x, y), z\rangle_e &= \langle [x, y], z\rangle_e - \langle \mathrm{ad}_y^*(x), z\rangle_e - \langle \mathrm{ad}_x^*(y), z\rangle_e.\end{aligned}$$

Definition 4.7 (Structure constants)**.** Let G be a Lie group and $g = \langle \cdot, \cdot\rangle$ be a left-invariant metric on G. Let $(e_1, \dots, e_d)$ be an orthonormal basis of $\mathfrak{g}$ for g. Define the *structure constants* as

$$C_{ij}^k = \langle [e_i, e_j], e_k\rangle.$$

Example 4.12: Structure constants on SO(3)

An orthonormal basis of $\mathfrak{so}(3)$ endowed with the Frobenius metric is

$$a_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \qquad a_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \qquad a_3 = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

.

And as the Lie bracket is the matrix commutator, we can compute the structure constants to obtain

$$C_{ij}^k = \frac{1}{\sqrt{2}} \text{ if } ijk \text{ is a direct cycle of } \{1,2,3\},$$

and 0 otherwise.

Example 4.13: Structure constants on SE(3)

We define a left-invariant metric on the special Euclidean group by defining an inner product in its Lie algebra. Let the metric matrix at the identity be diagonal: $g = \mathrm{diag}(1,1,1,\beta,1,1)$ for some $\beta > 0$, an anisotropy parameter. For $\beta = 1$, this metric coincides with the Frobenius metric. An orthonormal basis of the Lie algebra $\mathfrak{se}(3)$ for this metric is

$$\begin{array}{ccc} e_1 = \begin{pmatrix} a_1 & 0 \\ 0 & 0 \end{pmatrix} & e_2 = \begin{pmatrix} a_2 & 0 \\ 0 & 0 \end{pmatrix} & e_3 = \begin{pmatrix} a_3 & 0 \\ 0 & 0 \end{pmatrix} \\ e_4 = \frac{1}{\sqrt{\beta}}\begin{pmatrix} 0 & \epsilon_1 \\ 0 & 0 \end{pmatrix} & e_5 = \begin{pmatrix} 0 & \epsilon_2 \\ 0 & 0 \end{pmatrix} & e_6 = \begin{pmatrix} 0 & \epsilon_3 \\ 0 & 0 \end{pmatrix}. \end{array}$$

where (a_1, a_2, a_3) is the basis of $\mathfrak{so}(3)$ defined above, and $(\epsilon_1, \epsilon_2, \epsilon_3)$ is the canonical basis of $\mathbb{R}^3$. As the Lie bracket is the usual matrix commutator, it is straightforward to compute

$$C_{ij}^k = \frac{1}{\sqrt{2}} \text{ if } ijk \text{ is a direct cycle of } \{1,2,3\}; \tag{4.6}$$

$$C_{15}^6 = -C_{16}^5 = -\sqrt{\beta} C_{24}^6 = \frac{1}{\sqrt{\beta}} C_{26}^4 = \sqrt{\beta} C_{34}^5 = -\frac{1}{\sqrt{\beta}} C_{35}^4 = \frac{1}{\sqrt{2}}. \tag{4.7}$$

and all others that cannot be deduced by skew-symmetry of the bracket are equal to 0.

The structure constants allow to compute ad^* and formula (4.4) in practice:

$$\alpha(e_i, e_j) = \nabla_{e_i} e_j = \frac{1}{2} \sum_k (C_{ij}^k - C_{jk}^i + C_{ki}^j) e_k.$$

Note however that eq. (4.4) gives the connection for left-invariant vector fields only. We will now generalize to any vector field defined along a smooth curve on G, using the left-invariant basis $(\tilde{e}_1, \dots, \tilde{e}_n)$.

Let $\gamma : [0,1] \to G$ be a smooth curve, and Y a vector field defined along γ. Write $Y = h^i \tilde{e}_i$, $\dot{\gamma} = f^i \tilde{e}_i$. let us also define the *left-angular velocities* $\omega = dL_\gamma^{-1}\dot{\gamma} = f^i \circ \gamma e_i \in \mathfrak{g}$ and $\zeta = dL_\gamma^{-1} Y = h^j \circ \gamma e_j \in \mathfrak{g}$. Then the covariant derivative of Y along γ is

$$\begin{aligned} \nabla_{\dot{\gamma}(t)} Y &= (f^i \circ \gamma)(t) \nabla_{\tilde{e}_i} \big(h^j \tilde{e}_j\big) \\ &= (f^i \circ \gamma)(t) \tilde{e}_i(h^j) \tilde{e}_j + (f^i \circ \gamma)(t)(h^j \circ \gamma)(t)(\nabla_{\tilde{e}_i} \tilde{e}_j)_{\gamma(t)} \\ dL_{\gamma(t)}^{-1} \nabla_{\dot{\gamma}(t)} Y &= (f^i \circ \gamma)(t) \tilde{e}_i(h^j) e_j + (f^i \circ \gamma)(t)(h^j \circ \gamma)(t) dL_{\gamma(t)}^{-1} (\nabla_{\tilde{e}_i} \tilde{e}_j)_{\gamma(t)} \\ &= (f^i \circ \gamma)(t) \tilde{e}_i(h^j) e_j + (f^i \circ \gamma)(t)(h^j \circ \gamma)(t) \nabla_{e_i} e_j \end{aligned}$$

where the Leibniz formula and the invariance of the connection is used in $(\nabla_{\tilde{e}_i} \tilde{e}_j) = dL_{\gamma(t)} \nabla_{e_i} e_j$. Therefore, for $k = 1..n$

$$\begin{aligned} \langle dL_{\gamma(t)}^{-1} \nabla_{\dot{\gamma}(t)} Y, e_k \rangle = &(f^i \circ \gamma)(t) \tilde{e}_i(h^j) \langle e_j, e_k \rangle \\ &+ (f^i \circ \gamma)(t)(h^j \circ \gamma)(t) \langle \nabla_{e_i} e_j, e_k \rangle \end{aligned}$$

but on the one hand

$$\begin{aligned}
\zeta(t) &= (h^j \circ \gamma)(t) e_j \\
\dot{\zeta}(t) &= (h^j \circ \gamma)'(t) e_j = dh^j_{\gamma(t)} \dot{\gamma}(t) e_j \\
&= dh^j_{\gamma(t)} \Big((f^i \circ \gamma)(t) \tilde{e}_{i,\gamma(t)} \Big) e_j \\
&= (f^i \circ \gamma)(t) d_{\gamma(t)} h^j \tilde{e}_{i,\gamma(t)} e_j \\
&= (f^i \circ \gamma)(t) \tilde{e}_i (h^j)_{\gamma(t)} e_j
\end{aligned}$$

and on the other hand, using equation (4.4):

$$\begin{aligned}
(f^i \circ \gamma)(h^j \circ \gamma) \langle \nabla_{e_i} e_j, e_k \rangle &= (f^i \circ \gamma)(h^j \circ \gamma) \Big(\langle [e_i, e_j], e_k \rangle - \langle \mathrm{ad}^*_{e_i} e_j, e_k \rangle - \langle \mathrm{ad}^*_{e_j} e_i, e_k \rangle \Big) \\
&= \frac{1}{2} (f^i \circ \gamma)(h^j \circ \gamma) \Big(\langle [e_i, e_j], e_k \rangle - \langle [e_j, e_k], e_i \rangle - \langle [e_i, e_k], e_j \rangle \Big) \\
&= \frac{1}{2} (\langle [(f^i \circ \gamma) e_i, (h^j \circ \gamma) e_j], e_k \rangle \\
&\quad - \langle [(h^j \circ \gamma) e_j, e_k], (f^i \circ \gamma) e_i \rangle \\
&\quad - \langle [(f^i \circ \gamma) e_i, e_k], (h^j \circ \gamma) e_j \rangle) \\
&= \langle \frac{1}{2} \left([\omega, \zeta] - \mathrm{ad}^*_{\omega} \zeta - \mathrm{ad}^*_{\zeta} \omega \right), e_k \rangle = \langle \alpha(\omega, \zeta), e_k \rangle.
\end{aligned}$$

Thus, we obtain an algebraic expression for the covariant derivative of any vector field Y along a smooth curve γ. It is the main ingredient of this section.

Lemma 4.1. *Let G be a Lie group and ∇ be the Levi-Civita connection of a left-invariant metric on G. Let γ be a smooth curve on G, and Y a vector field defined along γ. Consider the left-angular velocities $\omega = dL_\gamma^{-1} \dot{\gamma}$ and $\zeta = dL_\gamma^{-1} Y$. Then*

$$dL^{-1}_{\gamma(t)} \nabla_{\dot{\gamma}(t)} Y(t) = \dot{\zeta}(t) + \alpha(\omega(t), \zeta(t)) \tag{4.8}$$

A similar expression can be found in Arnold 1966; Gay-Balmaz, Holm, et al. 2012. As all the variables of the right-hand side are defined in $\mathfrak{g}$, they can be computed with matrix operations and an orthonormal basis.

We now focus on two particular cases of (4.8) to derive the equations of geodesics and of parallel transport along a curve.

4.3.3 Geodesic equation

The first particular case is for $Y(t) = \dot{\gamma}(t)$. It is then straightforward to deduce from equation (4.8) the Euler-Poincaré equation for a geodesic curve (e.g Kolev 2004; Cendra, Holm, et al. 1998). Indeed in this case, recall that $\omega = dL_\gamma^{-1} \dot{\gamma}$ is the left-angular velocity, $\zeta = \omega$ and $\alpha(\omega, \omega) = -\mathrm{ad}^*_\omega(\omega)$. Hence γ is a geodesic if and only if $dL^{-1}_{\gamma(t)} \nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0$ i.e. setting the left-hand side of equation (4.8) to 0. We obtain

$$\begin{cases} \dot{\gamma}(t) &= dL_{\gamma(t)} \omega(t) \\ \dot{\omega}(t) &= \mathrm{ad}^*_{\omega(t)} \omega(t). \end{cases} \tag{4.9}$$

Remark 4.6.

- *A similar treatment of a right-invariant metric is straightforward. Indeed, in this case the Lie bracket of right-invariant vector fields is the opposite of the adjoint representation of the Lie algebra, therefore the expressions are all the same with $-\alpha$ instead of α.*

- *One can show that the metric is bi-invariant if and only if the adjoint map is skew-symmetric (see Xavier Pennec and Arsigny 2013 or Gallier and Quaintance 2020, Prop. 20.7). In this case* $\mathrm{ad}_\omega^*(\omega) = 0$ *and equation* (4.9) *coincides with the equation of one-parameter subgroups on* G*.*

In the class `InvariantMetric` of geomstats, the `geodesic_equation` method is modified to implement equation (4.9), and the integrators of Section 3.3 (page 34) are used to compute the exponential map.

Example 4.14: Geodesics on SE(2)

We use the same visualization as in Example 4.9 (page 48) to plot the geodesics of a left-invariant metric on $SE(2)$. We first compare the left and right-invariant metrics with same inner-product at the identity, and compare their geodesics to one parameter subgroups. These curves are shown on Figure 13.

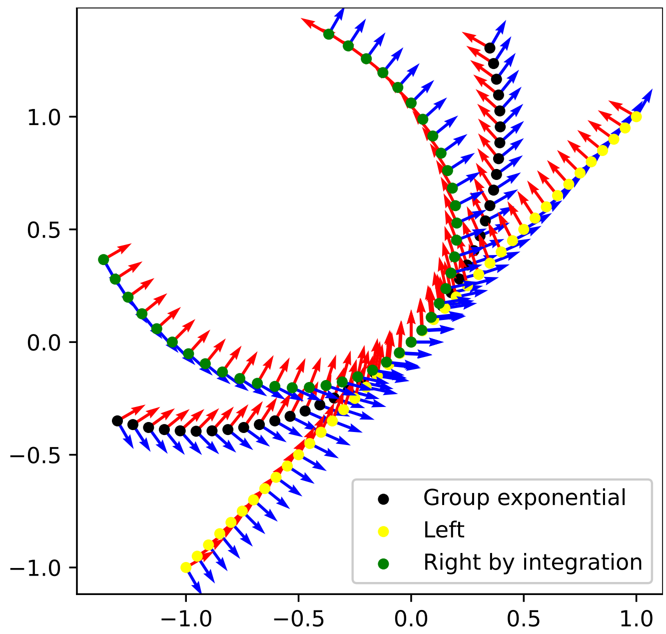


Figure 13: Visualization of geodesics and one parameter subgroups on $SE(2)$.

Next, we define the metric as in Example 4.13, by the bilinear form defined in $\mathfrak{se}(2)$ by the matrix $g = \mathrm{diag}(1, \beta, 1)$ for some $\beta > 0$, an anisotropy parameter, and extended by left-invariance.

For $\beta = 1$, the metric coincides with the Frobenius metric, and there is no interaction between the rotation and translation part. This metric is thus isomorphic to the direct product metric of the bi-invariant metric on $SO(2)$ and the usual inner product of $\mathbb{R}^2$. The geodesics are thus straight-lines on the translation part, and rotations with constant speed (as one-parameter subgroups) on the rotation part.

When $\beta \neq 1$, the rotation part remains the same, but there is a direction towards which less movement occurs. This is shown on Figure 14.

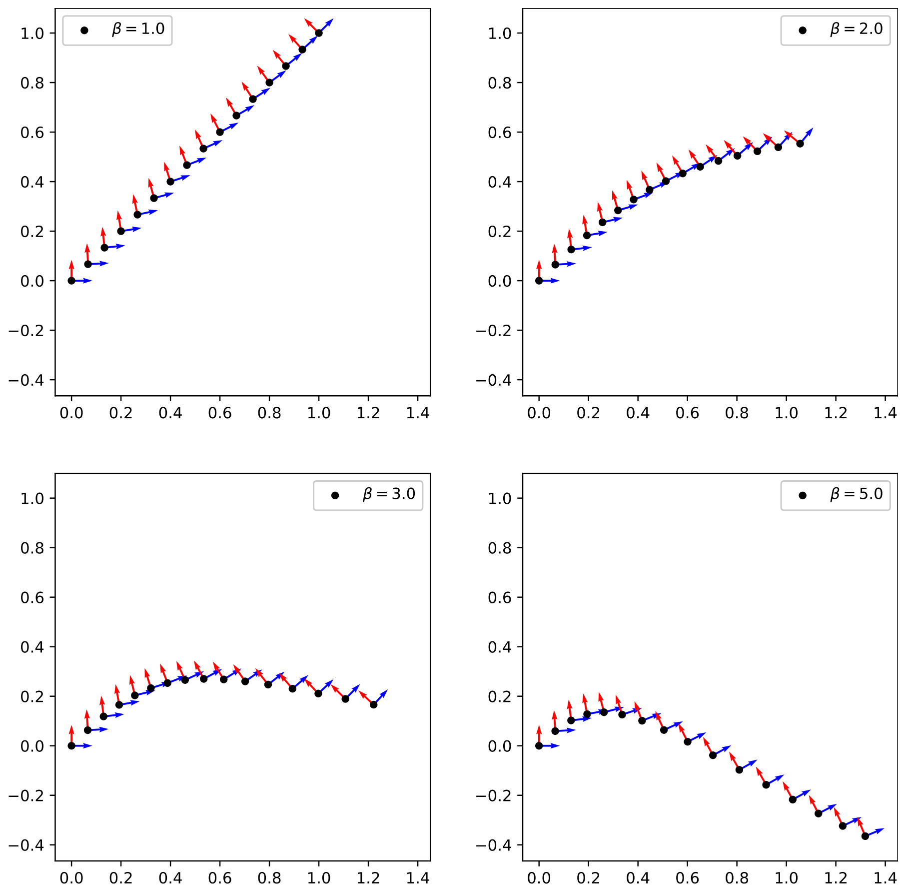


Figure 14: Visualization of geodesics on $SE(2)$ with the same initial conditions and different values of β.

4.3.4 Reduced parallel transport equation

The second case is for a vector field Y that is parallel along the curve γ, that is, $\forall t, \nabla_{\dot\gamma(t)} Y(t) = 0$. Similarly to the geodesic equation, we deduce from equation (4.8) the parallel transport equation expressed in the Lie algebra. To the best of our knowledge, this formulation of the parallel transport on Lie groups with an invariant metric is original.

Theorem 4.4. *Let γ be a smooth curve on G. The vector Y is parallel along γ if and only if it is solution to*

$$\begin{cases} \omega(t) & = dL^{-1}_{\gamma(t)} \dot\gamma(t) \\ Y(t) & = dL_{\gamma(t)} \zeta(t) \\ \dot\zeta(t) & = -\alpha(\omega(t), \zeta(t)) \end{cases} \tag{4.10}$$

Note that in order to parallel transport along a geodesic curve, equations (4.9) and (4.10) are solved jointly. We add the corresponding method to the `InvariantMetric` class of geomstats.

```
def geodesic_equation(self, state, _time):
    """Compute the right-hand side of the geodesic equation."""
```

```python
    sign = 1. if self.left_or_right == 'left' else -1.
    basis = self.normal_basis(self.lie_algebra.basis)

    point, vector = state
    velocity = self.group.tangent_translation_map(
        point, left_or_right=self.left_or_right)(vector)
    coefficients = gs.array([self.structure_constant(
        vector, basis_vector, vector) for basis_vector in basis])
    acceleration = gs.einsum('i...,ijk->...jk', coefficients, basis)
    return gs.stack([velocity, sign * acceleration])

def parallel_transport(
        self, tangent_vec_a, tangent_vec_b, base_point, n_steps=10,
        step='rk4', **kwargs):
    """Compute the parallel transport of a tangent vector along a geodesic."""
    group = self.group
    translation_map = group.tangent_translation_map(
        base_point,
        left_or_right=self.left_or_right, inverse=True)
    left_angular_vel_a = group.to_tangent(translation_map(tangent_vec_a))
    left_angular_vel_b = group.to_tangent(translation_map(tangent_vec_b))

    def acceleration(state, time):
        point, omega, zeta = state
        gam_dot, omega_dot = self.geodesic_equation(state[:2], time)
        zeta_dot = - self.connection_at_identity(omega, zeta)
        return gs.stack([gam_dot, omega_dot, zeta_dot])

    if (base_point.ndim == 2 or base_point.shape[0] == 1) and \
            (tangent_vec_a.ndim == 3 or tangent_vec_b.ndim == 3):
        n_sample = tangent_vec_a.shape[0] if tangent_vec_a.ndim == 3 else\
            tangent_vec_b.shape[0]
        base_point = gs.stack([base_point] * n_sample)

    initial_state = gs.stack([
        base_point, left_angular_vel_b, left_angular_vel_a])
    flow = integrate(
        acceleration, initial_state, n_steps=n_steps, step=step, **kwargs)
    gamma, gamma_dot, zeta_t = flow[-1]
    transported = group.tangent_translation_map(
        gamma, left_or_right=self.left_or_right, inverse=False)(zeta_t)
    return transported
```

Example 4.15: Parallel transport in $SE(3)$

We exemplify the use of the reduced parallel transport equation on $SE(3)$ endowed with an anisotropic left-invariant metric, as in Examples 4.13 and 4.14. The metric is defined by the matrix $g = \mathrm{diag}(1, 1, 1, \beta, 1, 1)$ for some $\beta > 0$ at the identity and extended by left-invariance.

We randomly generate a point in $x \in SE(3)$ and two tangent vectors $v, w \in T_x SE(3)$, and transport the vector v along the geodesic $t \mapsto \mathrm{Exp}_x(tw)$.

When $\beta = 1$, it coincides with the product metric between the bi-invariant metric of $SO(3)$ and of $\mathbb{R}^3$, so the geodesics and parallel transport are known in closed form. For $\beta > 1$, the curvature grows away from 0 (this is proved in Appendix B), and so does its covariant derivative, so that parallel transport becomes more complex to compute. we use $n = 1100$ steps in the discretization of $[0, 1]$ for integration scheme to compute a reference value that is then used to measure the error of the method for $10 \leq n \leq 1000$. We use different values of β but keep the same initial vectors regardless of the value of β.

The results are plotted in a loglog plot, with the error shown with respect to the number of steps used to integrate equations (4.9) and (4.10), with an $RK2$ or

$RK4$ scheme. As expected, the speed of convergence depends on the order of the scheme used: quadratic speed is reached for $RK2$ and speed of order 4 for $RK4$. The method is very stable for large n, and shows little sensibility to a change of metric (i.e. when β grows).

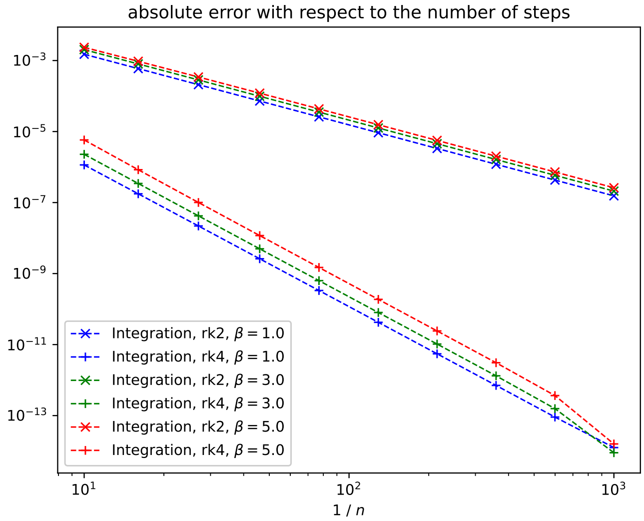


Figure 15: Norm of the absolute error represented with respect to the number of steps.

4.3.5 Curvature

By definition of the left-invariance of the metric, for any $g \in G$, the left-translation map L_g is an isometry. It is thus sufficient to compute the curvature tensor at the identity, that is, in the Lie algebra $\mathfrak{g}$, and to map it to any other point. Using an orthonormal basis and formula (4.4), this reduces to simple algebra. The covariant derivative of the curvature tensor can also be computed using Leibniz rule, for any $u, v, w, z \in \mathfrak{g}$:

$$(\nabla_z R)(u,v)w = \nabla_z\big(R(u,v)w\big) - R(\nabla_z u, v)w - R(u, \nabla_z v)w - R(u,v)\nabla_z w \qquad (4.11)$$

Of course this is also valid for right-invariant metrics. This is implemented in geomstats in the class `InvariantMetric` as follows.

```
def connection_at_identity(self, tangent_vec_a, tangent_vec_b):
    """Compute the Levi-Civita connection at identity."""
    sign = 1. if self.left_or_right == 'left' else -1.
    return sign / 2 * (Matrices.bracket(tangent_vec_a, tangent_vec_b)
                       - self.dual_adjoint(tangent_vec_a, tangent_vec_b)
                       - self.dual_adjoint(tangent_vec_b, tangent_vec_a))

def curvature_at_identity(
        self, tangent_vec_a, tangent_vec_b, tangent_vec_c):
    """Compute the curvature at identity."""
    bracket = Matrices.bracket(tangent_vec_a, tangent_vec_b)
    bracket_term = self.connection_at_identity(bracket, tangent_vec_c)

    left_term = self.connection_at_identity(
        tangent_vec_a, self.connection_at_identity(
            tangent_vec_b, tangent_vec_c))

    right_term = self.connection_at_identity(
```

```python
            tangent_vec_b, self.connection_at_identity(
                tangent_vec_a, tangent_vec_c))

        return left_term - right_term - bracket_term

    def sectional_curvature_at_identity(self, tangent_vec_a, tangent_vec_b):
        """Compute the sectional curvature at identity."""
        curvature = self.curvature_at_identity(
            tangent_vec_a, tangent_vec_b, tangent_vec_b)
        num = self.inner_product(curvature, tangent_vec_a)
        denom = (
                self.squared_norm(tangent_vec_a)
                * self.squared_norm(tangent_vec_a)
                - self.inner_product(tangent_vec_a, tangent_vec_b) ** 2)
        condition = gs.isclose(denom, 0.)
        denom = gs.where(condition, 1., denom)
        return gs.where(~condition, num / denom, 0.)

    def curvature_derivative_at_identity(
            self, tangent_vec_a, tangent_vec_b, tangent_vec_c, tangent_vec_d):
        """Compute the covariant derivative of the curvature at identity."""
        first_term = self.connection_at_identity(
            tangent_vec_a,
            self.curvature_at_identity(
                tangent_vec_b, tangent_vec_c, tangent_vec_d))

        second_term = self.curvature_at_identity(
            self.connection_at_identity(tangent_vec_a, tangent_vec_b),
            tangent_vec_c,
            tangent_vec_d)

        third_term = self.curvature_at_identity(
            tangent_vec_b,
            self.connection_at_identity(tangent_vec_a, tangent_vec_c),
            tangent_vec_d)

        fourth_term = self.curvature_at_identity(
            tangent_vec_b,
            tangent_vec_c,
            self.connection_at_identity(tangent_vec_a, tangent_vec_d))

        return first_term - second_term - third_term - fourth_term
```

If the metric is bi-invariant, it is well known that these formulas greatly simplify (Lafontaine, Gallot, et al. 2004, Proposition 3 .17) or (Gallier and Quaintance 2020, Proposition 20.19).

Proposition 4.5. *If G is a Lie group equipped with a bi-invariant metric, and if $X, Y, Z, T \in \mathcal{L}(G)$, then*

$$\langle R(X,Y)Z, T\rangle = \frac{1}{4}\langle [X,Y],[Z,T]\rangle \tag{4.12}$$

In particular, G has non-negative sectional curvature.

We now turn to the case of submersions defined by the action of a Lie group on a manifold, and the properties of the induced quotient metric.

4.4 Group action and homogeneous spaces

To conclude this section, we define the notion of group action, that will be fundamental in Section 5. It allows to model groups as generators of transformations of any manifold. In Example 2.5 (page 9), the action of $SO(m)$ on $M_{m,k}(\mathbb{R})$ is in fact the main factor to study the geometry of the quotient space $S\Sigma_m^k$.

Definition 4.8 (Group action)**.** Given a set M and a group G, a *left-action* of G on M is a function $\triangleright : G \times M \to M$, such that:

- $\forall g, h \in G, \forall x \in M, \quad g \triangleright (h \triangleright x) = (gh) \triangleright x,$
- $\forall x \in M, \quad e \triangleright x = x.$

If furthermore M is a smooth manifold and G is a Lie group, we say that this action is smooth if the map $\triangleright$ is smooth from the product manifold $G \times M$ to M. In this case, for any $g \in G$, $x \mapsto g \triangleright x$ is a diffeomorphism of M, whose inverse is $x \mapsto g^{-1} \triangleright x$. We call this map *left translation* by g and write it L_g by analogy with the group law.

Remark 4.7. *Note that the map $g \mapsto L_g$ is a group homomorphism between G and* Diff(M)*. In fact, if this mapping is injective, we say that the action is faithful and we can see G as an immersed subgroup of* Diff(M)*.*

We say that an action is

- *free* if for all $g \in G$ and $x \in M$, if $g \triangleright x = x$, then $g = e$;
- *faithful* if for all $g \in G$, if $\forall x \in M, g \triangleright x = x$ then $g = e$;
- *transitive* if for all $x, y \in M$, there exists $g \in G$ such that $y = g \triangleright x$;
- *proper* if for all compact set $K \subset M$, the set $\{g \in G \mid K \cap gK \neq \emptyset\}$ is compact. This is always the case if G is compact.

Further, The define for any $x \in M$ the *orbit* of x as $[x] = G \triangleright x = \{g \triangleright x \mid g \in G\}$. As in Example 2.5 (page9), the orbits are the equivalence classes of the following relation

$$x \sim y \iff \exists g \in G, y = g \triangleright x \iff y \in g \triangleright x.$$

Note that a free action means that $G_x = e$ for all $x \in M$. In contrast, if an action is faithful, then the map $x \mapsto g \triangleright x$ is different from the identity of M for all $g \in G$. These notions play an essential role in determining the geometry of M. For example, we shall see in Kendall shape spaces that the action of $SO(n)$ is free if we remove certain points, that are otherwise considered as singularities.

Definition 4.9 (Quotient space)**.** The set of orbits is denoted M/G and is called *quotient* of M by G. This set is obtained by identifying all the points in an orbit. Define the *canonical projection*

$$\pi : \left\{ \begin{array}{ccc} M & \longrightarrow & M/G \\ x & \longmapsto & [x] \end{array} \right. \tag{4.13}$$

We have the following sufficient conditions on the action to ensure that a quotient space is indeed a smooth manifold (Lee 2003, Theorem 7.10).

Theorem 4.5 (Quotient manifold)**.** *Let M be a smooth manifold, G a Lie group with a smooth left-action on M that is free and proper. Then there exist a unique differential structure on M/G such that M/G is a smooth manifold and the canonical projection (4.13) is a submersion.*

Note that M/G is equipped with the quotient topology, and the differential structure on M/G can be derived by taking charts of M that are *adapted* to the group action in the sense that the open sets intersect with only one orbit. See Lee 2003 for more details. Free and proper actions occur in many manifolds considered until here, e.g. the Stiefel manifold (Example 2.4 page 8).

We now focus on a particular case of the action of a subgroup of G on G. At any $x \in M$, we can define the *isotropy* subgroup, or *stabilizer* of x as

$$G_x = \{g \in G \mid g \triangleright x = x\}.$$

Then G_x is a Lie subgroup of G. We consider the action of G_x on G, or more generally of a Lie subgroup H of G, obviously defined by the group law. We have (Gallier and Quaintance 2020, Corollary 22.10):

Theorem 4.6. *The action of a Lie subgroup H of a Lie group G on G is free and proper.*

Theorem 4.5 thus applies to the right action $(g,h) \in G \times H \mapsto gh \in G$ and G/H is a smooth manifold such that $\pi : G \to G/H$ is a submersion. The orbits of this right action are the left cosets $\{\mathrm{gH} \mid g \in G\}$ Note that G acts on G/H by $g_1 \triangleright g_2 H = g_1 g_2 H$, and it is clear that this action is transitive. In fact all transitive left actions yield to a quotient space of orbits from a right action.

Definition 4.10 (Homogeneous space)**.** We say that M is homogeneous if there exists a smooth transitive action of a Lie group G on M.

Remark 4.8. *Note that there might be several such Lie groups with a transitive action on a given homogeneous space M.*

All homogenous spaces correspond (up to a diffeomorphism) to quotient spaces of the form G/H, as stated in the following (Gallier and Quaintance 2020, Theorem 22.13).

Theorem 4.7. *Let G be a connected Lie group acting smoothly and transitively on a smooth manifold M, so that M is homogeneous. Then for any $x \in M$, writing $H = G_x$, the map*

$$\begin{cases} G/H & \longrightarrow & M \\ gH & \longmapsto & g \triangleright x \end{cases}$$

is a diffeomorphism, so that $M \simeq G/H$.

Note that the choice of reference point x above does not matter, as for two different choices x and y, the action is transitive so there exists $g \in G$ such that $y = g \triangleright x$, and the stabilizers G_x and $G_y = G_{g \triangleright x} = gG_x g^{-1}$ are conjugate, hence isomorphic.

We exemplify all these notions with the sphere, the Stiefel manifold and the manifold of symmetric positive definite (SPD) matrices, although other manifolds in geomstats such as hyperbolic spaces and the Grassmannian are also homogeneous.

Example 4.16: Hypersphere as a homogenous space

The hypersphere embedded in $\mathbb{R}^{d+1}$ can be seen as a homogeneous space by considering the action of the special orthogonal group $SO(d+1)$ of the embedding space. For $R \in SO(d+1), x \in S^d$:

$$R \triangleright x = Rx \in S^d \tag{4.14}$$

Indeed, the sphere is stable by the action of rotation matrices, and this action is transitive. Consider a pole $x_0 = (1, 0, \dots, 0) \in S^d$. Its stabilizer is the set of rotations whose axis is x_0, i.e.

$$H = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & R \end{pmatrix} \mid R \in SO(d) \right\} \simeq SO(d).$$

Then by Theorem 4.7, $S^d = SO(d+1)/SO(d)$. Intuitively this corresponds to identifying all the rotations that share the same axis where rotation axes are described by a unit vector, i.e., a point on the sphere.

Example 4.17: Stiefel manifold as a homogenous space

Recall that the Stiefel manifold $St(k, n)$ is the set of orthonormal k-frames of $\mathbb{R}^n$. The special orthogonal group $SO(n)$ thus acts on $St(k, n)$ by

$$R \triangleright (u_1, \ldots, u_k) = (Ru_1, \ldots, Ru_k)$$

It is clear that this action is transitive. Furthermore, as a frame can be represented by an orthogonal matrix of size $n \times k$, the above action corresponds to matrix multiplication and is thus smooth.

For any $U \in St(k, n)$, the stabilizer of U is

$$H = \left\{ \begin{pmatrix} I_k & 0 \\ 0 & R \end{pmatrix} \mid R \in SO(n-k) \right\} \simeq SO(n-k).$$

Then by Theorem 4.7, $St(k, n) = SO(n)/SO(n-k)$. Intuitively, this corresponds to the fact that a k-frame can be completed into an orthonormal basis of $\mathbb{R}^n$, i.e. a matrix in $SO(n)$, and matrices in H map one such completion to all the other completions. Thus quotienting $SO(n)$ by $SO(n-k)$ amounts to identifying the orthonormal bases that agree on the first k vectors.

Example 4.18: Grassmann manifold as a homogenous space

Define the Grassmann manifold $Gr(n, k)$ as the set of subspaces of $\mathbb{R}^n$ of dimension k. On any k-dimensional subspace $\mathcal{U}$ is defined a unique orthogonal projector, i.e. a linear map p defined in $\mathbb{R}^n$ for which $p \circ p = p$ and $\text{Im}(p) = \mathcal{U}$ and $\ker(p) = \mathcal{U}^\perp$. Any such projector is represented by a symmetric matrix P of size n, rank k and such that $P^2 = P$. We thus adopt the representation of $Gr(n, k)$ as the set

$$Gr(n, k) = \left\{ P \in \text{Sym}(n) \mid P^2 = P \ \text{ and } \ \text{rank}(P) = k \right\}.$$

Intuitively, any k dimensional subspace can be represented by an equivalence class of orthonormal bases, so that the Grassmannian is a quotient space of the orthogonal group $O(n)$. To consider a connected group, let $G = SO(n)$, and let the action of G on $Gr(k, n)$ correspond to a change of basis: $Q \triangleright P = QPQ^\top$. It is clear that any rank-k projector can be represented by a matrix of the form

$$P_k = \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix}.$$

This means exactly that the action is transitive, and $Gr(k, n)$ is the orbit of P_k. Let H be its stabilizer, i.e. the set

$$H = \left\{ \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \mid Q \in SO(k), R \in SO(n-k) \right\} \simeq SO(k) \times SO(n-k).$$

$Gr(k, n) = SO(n)/(SO(k) \times SO(n-k))$ is therefore a homogeneous space with canonical projection $\pi : Q \in SO(n) \mapsto QP_kQ^\top$ and dimension $k(n-k)$. Of course this manifold also be described as a quotient of $St(k, n)$ by $SO(p)$, where the action is by right multiplication and the projection is identified with $U \in St(k, n) \mapsto UU^\top \in Gr(k, n)$. This quotient can be more practical than that of $SO(n)$ when $n \gg k$. The Grassmann manifold is widely used in applications both in numerical problems such as low-rank matrix decomposition or optimization, and in higher-level applications in machine learning, computer vision and image processing. We refer to Bendokat, Zimmermann, et al. 2020 for a very complete exposition of this manifold.

Example 4.19: SPD matrices as a homogenous space

Recall that a symmetric matrix Σ is *positive definite* if it is invertible and $\forall x \in \mathbb{R}^n, x^\top \Sigma x \geq 0$. The set $SPD(n)$ of such matrices is an open set of the vector space $\mathrm{Sym}(n)$ of symmetric matrices, hence it is an embedded manifold and the canonical immersion $\mathrm{id} : SPD(n) \to \mathrm{Sym}(n)$ defines a global chart. Furthermore, for any $\Sigma \in SPD(n)$ the tangent space $T_\Sigma SPD(n)$ is identified with $\mathrm{Sym}(n)$.

Now, define the action of $GL(n)$ on $SPD(n)$ by:

$$\triangleright : (A, \Sigma) \in GL(n) \times SPD(n) \longmapsto A \Sigma A^\top.$$

This action is sometimes called action by *congruence*. It is smooth and transitive. Indeed, let $\Sigma = PDP^\top, \Sigma' = Q\Delta Q^\top \in SPD(n)$ be two SPD matrices with their eigenvalue decomposition given by the spectral theorem, then $A = Q\Delta^{-1/2} D^{-1/2} P^\top$ is such that $A \triangleright \Sigma = \Sigma'$. The manifold of SPD matrices is thus a homogeneous space.

Finally, the isotropy group of the identity matrix I_n is the orthogonal group $O(n) \subset GL(n)$:

$$G_{I_n} = \{A \in GL(n) \mid A \triangleright I_n = I_n\} = \{A \in GL(n) \mid AA^\top = I_n\} = O(n).$$

It is indeed a closed subgroup of $GL(n)$, so its right action on $GL(n)$ is free and proper (Theorem 4.6), and by Theorem 4.7, $SPD(n)$ is isomorphic to the orbit space $GL(n)/O(n)$. The canonical projection $\pi : \left\{ \begin{array}{ccc} GL(n) & \longrightarrow & SPD(n) \\ A & \longmapsto & AA^\top \end{array} \right.$ is a submersion.

In geomstats, we implement a class for symmetric matrices and a class for the manifold of SPD matrices that inherits from `OpenSet`.

```
class SPDMatrices(OpenSet):
    """Class for the manifold of symmetric positive definite matrices."""

    def __init__(self, n):
        super().__init__(
            dim=int(n * (n + 1) / 2),
            ambient_space=SymmetricMatrices(n))
        self.n = n

    def belongs(self, mat, atol=gs.atol):
        """Check if a matrix is symmetric with positive eigenvalues."""
        is_symmetric = self.ambient_space.belongs(mat, atol)
        eigvalues = gs.linalg.eigh(mat, eigvals_only=True)
        is_positive = gs.all(eigvalues > 0, axis=-1)
        belongs = gs.logical_and(is_symmetric, is_positive)
        return belongs
```

As the covariance matrix of a multivariate random variable is an positive semi-definite matrix, this manifold is ubiquitous in applications such as signal processing, neuroscience, etc. (see Xavier Pennec, Sommer, et al. 2020, Chapter 3. and references therein), although some issues arise when the matrices are degenerate.

5 Metrics defined by invariance properties

As seen in the previous section, group actions play a key role in determining the geometric properties of a manifold. Designing operations that are either invariant or equivariant to a group action has been a fruitful source of improvement to deal with data lying on manifolds (e.g. Xavier Pennec, Fillard, et al. 2006), and more recently to generalize convolutional neural networks (Cohen, Geiger, et al. 2019, and references therein).

In this section, we focus on the implementation of Riemannian metrics that verify invariance properties on quotient manifolds, defining quotient metrics.

In the second case (Section 5.1), we propose a generic framework to implement quotient metrics, that are defined by a free and proper group action, and a metric that is invariant to this group action. This framework is exemplified in the case of Kendall shape spaces, and results in an efficient implementation that allows to compute parallel transport and curvature. This is, up to our knowledge, the first open-source Python implementation of such spaces. It was presented at the GSI 2021 conference in Guigui, Maignant, et al. 2021. Likewise, we implement the quotient metric on correlation matrices studied by Thanwerdas and Xavier Pennec 2021a.

This section thus gathers two contributions to the formulation and implementation of the parallel transport in the cases of invariant metrics and quotient metrics. Our original implementations allow to apply these results to numerous spaces. For the completeness of our exposure, we briefly present the particular cases of the invariant metrics on homogeneous and symmetric spaces in sections 5.2 and 5.3, and discuss the impact of these properties on the implementation.

5.1 Submersions and quotient metrics

A particularly interesting setting is when there exists a submersion $\pi : E \to M$ between two Riemannian manifolds E and M, such that π is compatible with the two metrics. In that case, most of the geometry of M can be deduced from the geometry of E and the knowledge of π. Such submersions arise for homogeneous spaces, and for quotients by a Lie group, hence we shall revisit some examples from the previous sections with this viewpoint. We first introduce the general notions, then focus on quotient metrics. Metrics on homogeneous spaces will be treated in Section 5.2.

Although the mathematical notions exposed in this section are well known since the works of O'Neill 1966 and Le and D. G. Kendall 1993, this work is original by the common implementation of these structures in geomstats. This implementation allowed computing parallel transport on Kendall shape spaces (Subsection 5.1.3), outperforming the state-of-the-art implementations, and are available on other quotient spaces with limited additional software development. The generic implementation is presented in Subsection 5.1.4, and constitutes one of the main contributions of this section.

To exemplify this section, we extensively use the work done in collaboration with Elodie Maignant on Kendall shape spaces, and presented at the GSI 2021 conference in Guigui, Maignant, et al. 2021. We also use the Bures-Wasserstein metric, with results from Bhatia, Jain, et al. 2019 and following Thanwerdas and Xavier Pennec 2021b.

5.1.1 Riemannian submersions

Throughout this section, let (E, g) and (M, h) be two Riemannian manifolds, and $\pi : E \to M$ be a submersion. Recall that this means that at every $p \in E$, $d\pi_p$ is surjective. We adopt the vocabulary of principal fiber bundles, referring to E as the *total* space, and to M as the *base* manifold. In particular, for every $x \in M$, $\pi^{-1}(x) \subset E$ is a submanifold in E usually called *fiber* above x, and the tangent space at any $p \in \pi^{-1}(x)$ is $T_p\pi^{-1}(x) = \ker d\pi_p$.

Definition 5.1 (Section). A *section* σ of $\pi : E \to M$ is a smooth map $\sigma : U \to E$ defined on an open set $U \subseteq M$ of M, such that $\forall x \in M, \quad \pi(\sigma(x)) = x$. If $U = M$, we say that σ is a *global* section.

A section allows to choose a point in a fiber above any x in a small open set. In the sequel, we assume that π always admits local sections. This is true for Riemannian submersions in particular by the inverse function theorem.

5.1.2 Quotient metric

Definition 5.2 (Horizontal - Vertical subspaces)**.** Let $x \in M$ and $p \in \pi^{-1}(x)$. The *vertical* subspace of T_pE is defined as $V_p = \ker d\pi_p$ and is the tangent space to the fiber through p. The *horizontal* subspace is its orthogonal complement in T_pE: $H_p = (V_p)^\perp$.

Any tangent vector $u \in T_pE$ can thus be decomposed into a vertical and a horizontal component, and we write ver and hor respectively the orthogonal projections on the vertical and horizontal subspace. Note that this decomposition depends on the metric on E by the definition of the orthogonal complement for the horizontal subspace. Furthermore, we say that u is horizontal if $u \in H_p$, and a curve $c : I \to E$ is called horizontal if at every time $t \in I$, the velocity of c is horizontal: $c'(t) \in H_{c(t)}$.

Example 5.1: SPD matrices with Bures-Wasserstein metric

Recall from Example 4.19 (page 61) that the map $A \in GL(n) \mapsto AA^\top \in SPD(n)$ is a smooth submersion. Its differential at any A is for any $H \in M_n(\mathbb{R})$:

$$d\pi_A H = AH^\top + HA^\top,$$

and its kernel is $\ker d\pi_A = \{H \mid HA^\top \in \mathrm{Skew}(n)\} = \mathrm{Skew}(n)A^{-\top}$. This defines the vertical subspace V_A.

Now, consider $GL(n)$ endowed with the restriction of the Frobenius metric. The orthogonal complement to V_A is

$$\begin{aligned} H_A &= \{H \in M_n(\mathbb{R}) \mid \forall B \in \mathrm{Skew}(n),\ \ \langle H, BA^{-\top}\rangle = 0\} \\ &= \{H \in M_n(\mathbb{R}) \mid \forall B \in \mathrm{Skew}(n),\ \ \mathrm{tr}(BA^{-\top}H^\top) = 0\} \\ &= \{H \in M_n(\mathbb{R}) \mid HA^{-1} \in \mathrm{Sym}(n)\} = \mathrm{Sym}(n)A. \end{aligned}$$

Note that in Section 5.3, we will consider the same manifolds and submersion, but instead endow $GL(n)$ with a left-invariant metric. This will result in another geometry on the manifold of SPD matrices.

Recall that as π is a submersion, $d\pi_p$ is a linear isomorphism between H_p and $\mathrm{Im}(\pi) = T_xM$. If this identification is additionally isometric, then geodesics and curvature on M will be deduced from E.

Definition 5.3 (Riemannian submersion)**.** Let $\pi : E \to M$ be a smooth submersion between two Riemannian manifolds. π is called a *Riemannian submersion* if for every $p \in E$, the map $d\pi_p$ is an isometry between the horizontal subspace H_p of T_pE and T_xM where $x = \pi(p)$.

If π is also surjective, this allows to identify tangent vectors of M to horizontal vectors of E.

Definition 5.4 (Horizontal lift)**.** Let $\pi : E \to M$ be a Riemannian submersion that is surjective onto M. Let X be a vector field on M, then the unique *horizontal lift* $\bar{X}$ in E is defined such that for every $x \in M$ and every $p \in \pi^{-1}(x)$,

$$d\pi_p \bar{X}_p = X_x.$$

If we have a local section σ, we speak about the *corresponding* horizontal lift such that $\forall (x, v) \in TM, \bar{v} \in H_{\sigma(x)} \subset T_{\sigma(x)}E$.

Example 5.2: Horizontal lift of SPD matrices

Following Example 5.1, it is known from linear algebra that the submersion $\pi : A \in GL(n) \mapsto AA^\top \in SPD(n)$ is surjective. To compute the horizontal lift of a tangent vector X at $\Sigma = AA^\top$, we seek a symmetric matrix S such that $d\pi_A(SA) = X$. This is true if and only if $AA^\top S + SAA^\top = X$, i.e. if $S \in \mathrm{Sym}(n)$ solves the Sylvester equation

$$\Sigma S + S\Sigma = X. \tag{5.1}$$

More precisely, using the eigenvalue decomposition of $\Sigma = PDP^\top$, S solves equation (5.1) iff

$$\begin{aligned} & \Sigma S + S\Sigma = X \\ \iff \quad & DP^\top SP + P^\top SPD = P^\top XP \\ \iff \quad & (P^\top SP)_{ij} = \frac{(P^\top XP)_{ij}}{d_i + d_j}. \end{aligned} \tag{5.2}$$

Define the map $S_\Sigma : \mathrm{Sym}(n) \to \mathrm{Sym}(n)$ that solves Sylvester equation using equation (5.2). It uniquely defines the horizontal lift $\bar{X}$ of X at A by $\bar{X} = S_{AA^\top}(X)A$.

Proposition 5.1. *Similarly, if $c : [0,1] \to M$ is a piecewise smooth curve in M, and $p \in \pi^{-1}(c(0))$, then there exists a unique curve $\bar{c} : [0,1] \to E$ such that $\pi \circ \bar{c} = c$ and $\bar{c}' \in H$. The curve $\bar{c}$ is called horizontal lift of c.*

Geodesics This leads to the main theorem of this section, that is usually attributed to O'Neill, and relates geodesics of M with horizontal geodesics of E (Gallier and Quaintance 2020, Proposition 17.8).

Theorem 5.1. *Let $\pi : E \to M$ be a Riemannian submersion between two Riemannian manifolds E and M.*

(1) If γ is a geodesic in E such that $c'(0)$ is horizontal, then c is horizontal, and its projection $\gamma = \pi \circ c$ is a geodesic in M of same length as c.

(2) For every $p \in E$, if γ is a geodesic in M such that $\gamma(0) = \pi(p)$, then there exists $\epsilon > 0$ such that there exists a unique horizontal lift c of the restriction of γ to $[-\epsilon, \epsilon]$ and c is a geodesic of E through p.

(3) If furthermore π is surjective, for any vector fields $X, Y \in \Gamma(TM)$, $\langle \bar{X}, \bar{Y} \rangle = \langle X, Y \rangle \circ \pi$ (where $\bar{\cdot}$ is the horizontal lift).

The first property of this theorem can be written by the commutative diagram below:

$$\begin{tikzcd} H_p \arrow[r, "\mathrm{Exp}_p"] \arrow[d, "d\pi_p"] & E \arrow[d, "\pi"] \\ T_x M \arrow[r, "\mathrm{Exp}_x"] & M \end{tikzcd}$$

where $p \in E$ and $x = \pi(p)$. Furthermore, as the restriction of $d\pi$ on horizontal spaces is an isomorphism, choosing a section σ of E and the corresponding horizontal lift $\bar{\cdot}$ allows to compute the exponential map of M from the one of E:

$$\forall x \in M, \forall v \in T_x M, \quad \mathrm{Exp}_x(v) = \pi \circ \mathrm{Exp}_{\sigma(x)}(\bar{v}) \tag{5.3}$$

Remark 5.1.

- *One can also show that the connection* $\nabla_{\bar{X}}\bar{Y}$ *on* E *verifies (see e.g. Lafontaine, Gallot, et al. 2004, Proposition 3.55)*

$$\nabla_{\bar{X}}\bar{Y} = \overline{\nabla_X Y} + \frac{1}{2}\operatorname{ver}[\bar{X},\bar{Y}].$$

  Consequently, one cannot hope to obtain a similar commutation rule between the parallel transport in E *and the one in* M*. Indeed, if* $\bar{Y}(t)$ *is the horizontal lift of a parallel vector field* $Y(t)$ *along a curve* γ *(whose horizontal lift is* $\bar{\gamma}$*), then* $\nabla_{\bar{\gamma}'}\bar{Y}$ *has a non-zero vertical component given above, which vanishes in the case of a geodesic. Computing parallel transport is thus not as straightforward as computing geodesics in this case.*

- *A Riemannian submersion shortens distances, i.e. for* $p, q \in E$*, and writing* d_E *and* d_M *the Riemannian distances on respectively* E *and* M*, we have*

$$d_M\big(\pi(p), \pi(q)\big) \leq d_E(p, q).$$

Example 5.3: Bures-Wasserstein geodesics

There is a unique metric that turns $\pi : A \in GL(n) \mapsto AA^\top \in SPD(n)$ into a Riemannian submersion between $GL(n)$ endowed with the Frobenius metric and $SPD(n)$. This metric is called the Bures-Wasserstein (BW) metric. let us apply Theorem 5.1 to compute the geodesics of this metric. Recall that $(GL(n), \text{Frobenius})$ is flat, so the geodesics are "straight" lines: $\operatorname{Exp}_A(tX) = A + tX$. Let $\Sigma = AA^\top \in SPD(n)$ and $X \in T_\Sigma SPD(n)$, then $\forall t > 0$

$$\begin{aligned}
\operatorname{Exp}_\Sigma(tX) &= \pi \circ \operatorname{Exp}_A(t\bar{X}) = \pi(A + tS_\Sigma(X)A) \\
&= (A + tS_\Sigma(X)A)(A + tS_\Sigma(X)A)^\top \\
&= AA^\top + t(S_\Sigma(X)\Sigma + \Sigma S_\Sigma(X)) + t^2 S_\Sigma(X)\Sigma S_\Sigma(X) \\
&= AA^\top + tX + t^2 S_\Sigma(X)\Sigma S_\Sigma(X)
\end{aligned}$$

Note that a geodesic hits the boundaries of the manifold when $-1/t$ is in an eigenvalue of $S_\Sigma(X)$. $(SPD(n), \text{BW})$ is therefore not complete. All the properties of this metric are derived using the equations of Riemannian submersions in Thanwerdas and Xavier Pennec 2021b; Thanwerdas 2022.

The relation between the connection of the total space and that of the base manifold also translates into a relation between the curvatures of each space.

Curvature O'Neill 1966 showed that the curvatures of the total space and of the base space were related, and that those of the base space could be computed using two fundamental tensors defined by the horizontal and vertical projections of the connection. We state here the main result, that shows that sectional curvature can only increase after a Riemannian submersion.

Theorem 5.2 (O'Neill)**.** *Let* $\pi : E \to M$ *be a Riemannian submersion, and* X, Y *be orthonormal vector fields on* M*, with horizontal lifts* $\bar{X}, \bar{Y}$*. Then*

$$\kappa(X, Y) = \kappa(\bar{X}, \bar{Y}) + \frac{3}{4}\|\operatorname{ver}[\bar{X}, \bar{Y}]\|^2. \tag{5.4}$$

Example 5.4: Curvature of the Bures-Wasserstein metric

As $GL(n)$ with the Frobenius metric is flat, O'Neill's theorem implies that the space of SPD matrices endowed with the Bures-Wasserstein metric has non-negative curvature.

We now apply these results to the special case of submersions defined as the canonical projection to a quotient space.

Metric Recall from Theorem 4.5 (page 58) that if G is a Lie group and E a smooth manifold such that G acts on E and the action is smooth, free and proper, then the canonical projection $\pi : x \in E \mapsto [x] \in M = E/G$ is a submersion.

In this case, the action of G allows to move in the fibers. More precisely, as the fibers are defined as orbits $G \triangleright p = \{g \triangleright p \mid g \in G\}$ for $p \in E$, the fibers are stable by the action of G, i.e. $\forall x \in M, \forall g \in G, \forall p \in \pi^{-1}(x), g \triangleright p \in \pi^{-1}(x)$, which can also be written $\pi(g \triangleright p) = \pi(p) = x$. Moreover, this action is transitive on fibers, i.e. for any $q \in E$ such that $\pi(q) = x$, there exists $g \in G$ such that $q = g \triangleright p$.

Definition 5.5 (Invariant metric)**.** A Riemannian metric on E is G-*invariant* if for any $g \in G$, $L_g : p \in E \mapsto g \triangleright p \in E$ is an isometry.

Now suppose that E is equipped with a G-invariant metric. Then one can define a metric on the quotient manifold $M = E/G$ such that π is a Riemannian submersion (Lafontaine, Gallot, et al. 2004, Proposition 2.28).

Proposition 5.2 (Quotient metric)**.** *Let E be a smooth manifold, G a Lie group acting smoothly, properly and freely on G, and $\langle \cdot, \cdot \rangle$ be a G-invariant metric on E . Let $\pi : E \to M$ be the canonical projection. Then there exists a unique Riemannian metric on M such that π is a Riemannian submersion. Let $x \in M$ and $p \in \pi^{-1}(x)$, for any $u, v \in T_xM$, it is given by*

$$\langle u, v \rangle_x \triangleq \langle \bar{u}, \bar{v} \rangle_p. \tag{5.5}$$

We refer to this metric on M as the quotient metric.

Thanks to the invariance of the metric on E, the quotient metric on M is well defined and does not depend on the choice of p in the fiber above x. The quotient metric is the unique metric on M such that π is a Riemannian submersion. Note that we do not use different notations for the two metrics for simplicity, but the subscripts indicate in which space it is defined: the character p is preferred for a point in the total space E and x in the quotient manifold M. Let d_E and d_M denote the Riemannian distance of respectively $(E, \langle \cdot, \cdot \rangle)$ and $(M, \langle \cdot, \cdot \rangle)$.

Proposition 5.3. *Let $p, q \in E$. We have the following relation*

$$d_M\big(\pi(p), \pi(q)\big) = \inf_{q' \in \pi^{-1}(\pi(q))} \{d_E(p, q')\} = \inf_{g \in G} \{d_E(p, gq)\}. \tag{5.6}$$

This leads to the following definition.

Definition 5.6 (Align)**.** Let $p, q \in E$, we say that q is *aligned* or *well-positioned* with p if $d_E(p, q) = d_M(\pi(p), \pi(q))$. If p and q are close enough, then there exists a unique q' aligned with p, and the geodesic between p and q is horizontal.

We define the *align* map ω on a sufficiently small subset of $D \subseteq E^2$, such that for any $(x, y) \in D$, $\omega(x, y) \in E$ is aligned with x. In general, the optimization problem corresponding to equation (5.6) must be solved in G to compute the align map. This procedure is often referred to as alignment, Procrustes analysis or registration in the literature. We implement a general optimization procedure using the `minimize` method of scipy, that implements a gradient descent. If the group and its Lie algebra are explicitly given, we optimize in the Lie algebra and use the group exponential map to map back to the group, then compute its action of a group element. A parameter represents the coefficient in a given basis of the Lie algebra, and `matrix_representation` maps these coefficient to the corresponding vector by linear combination.

```
def wrap(param):
    """Wrap a parameter vector to a group element and act on point."""
    algebra_elt = gs.array(param)
    algebra_elt = gs.cast(algebra_elt, dtype=base_point.dtype)
    algebra_elt = group.lie_algebra.matrix_representation(algebra_elt)
    group_elt = group.exp(algebra_elt)
    return group_action(point, group_elt)
```

If only the group action is given, we optimize without precautions on the group structure:

```
def wrap(param):
        """Wrap parameter vector to group element and act on point."""
        group_elt = gs.array(param)
        group_elt = gs.cast(group_elt, dtype=base_point.dtype)
        return group_action(group_eltr, point)
```

In both cases, automatic differentiation is used to compute the gradient of the objective function.

```
def align(self, point, base_point,
                max_iter=25, verbose=False, tol=gs.atol):
        """Align point to base_point by minimization."""
        ...  # definitions of wrap for different cases
        objective_with_grad = gs.autograd.value_and_grad(
                lambda param: self.ambient_metric.squared_dist(
                        wrap(param), base_point))

        init_param = gs.flatten(gs.random.rand(*max_shape))
        res = minimize(
                objective_with_grad, init_param, method='L-BFGS-B', jac=True,
                options={'disp': verbose, 'maxiter': max_iter}, tol=tol)

        return wrap(res.x)
```

This map is particularly useful because it allows to compute the Log map of the quotient space from the one of the total space. Indeed, choosing a section σ and the corresponding horizontal lift $\bar{\cdot}$, we can deduce from Theorem 5.1 the following relation:

$$\forall x, y \in M, \operatorname{Log}_x(y) = d\pi_{\sigma(x)} \operatorname{Log}_x\big(\omega(x, y)\big). \tag{5.7}$$

Example 5.5: Bures-Wasserstein distance

Continuing Example 5.2 where the submersion $A \mapsto AA^\top$ from $GL(n)$ to $SPD(n)$, and explaining why the metric of Example 5.3 is correctly defined, we define the quotient metric of $GL(n)$ with the Frobenius metric by the orthogonal group. Indeed, consider the right-action of $O(n)$ on $GL(n)$ by matrix multiplication. As shown in Example 4.19 (page 61), the canonical projection of $GL(n)/O(n)$ coincides with $\pi : A \mapsto AA^\top$.

The metric hereby defined on the manifold of SPD matrices is called the Bures-Wasserstein metric (Bhatia, Jain, et al. 2019). It arises naturally as the L_2-Wasserstein distance on multivariate centred normal random variables. It also corresponds to the Procrustes reflection size-and-shape metric (I. L. Dryden, Koloydenko, et al. 2009) taken on the square root SPD matrices. As $GL(n)$

is Euclidean under the Frobenius metric, the geodesics and curvature are known and can be projected to the quotient space. The minimization problem (5.6) can be solved in closed-form to obtain (Thanwerdas and Xavier Pennec 2021b; Thanwerdas 2022).

$$d(\Sigma, \Sigma')^2 = \mathrm{tr}(\Sigma) + \mathrm{tr}(\Sigma') - 2\,\mathrm{tr}((\Sigma\Sigma')^{1/2}).$$

In the next subsection, we focus on Kendall shape spaces, as most of the geometry of these spaces can be computed using the results of this section. Our implementation in geomstats is, to the best of our knowledge, the first open-source Python implementation of Kendall shape spaces, and allows efficient computation of parallel transport.

5.1.3 Application to Kendall shape spaces

The following subsection is inspired by Nava-Yazdani, Hege, et al. 2020 but constitutes an original contribution by putting all the code open source in our package geomstats, and by improving the implementation of parallel transport. It is a collaboration with Elodie Maignant and was presented at GSI 2021 in Guigui, Maignant, et al. 2021.

We revisit Example 2.5 (page 9) with the additional idea that the size of a set of landmarks may be filtered out to define a shape. The study of these spaces, including their mathematical structure, the properties of statistical distributions and estimation methods for shape data and their applications to many scientific fields goes back to the late 1970's with the works of Mardia, Bookstein and Kendall among others. For historical notes on this research areas and an introduction to the applications see I. L. Dryden and Mardia 2016, Preface and Section 1 .4. Recall that we consider the set of centred matrices of size $m \times k$ as the space of configurations (or landmarks). We assume that $m \geq 2$, and refer the reader to Le and D. G. Kendall 1993 for more details.

To further remove the effects of scaling, we restrict to non-zero x (i.e. at least two landmarks are different), and divide x by its Frobenius norm (written $\|\cdot\|$). This defines the pre-shape space

$$\mathcal{S}_m^k = \{x \in M(m,k) \mid \sum_{i=1}^k x_i = 0, \quad \|x\| = 1\},$$

which is identified with the hypersphere of dimension $m(k-1)-1$. The pre-shape space is therefore a differential manifold whose tangent space at any $x \in \mathcal{S}_m^k$ is given by

$$T_x\mathcal{S}_m^k = \{w \in M(m,k) \mid \sum_{i=1}^k w_i = 0, \ \mathrm{tr}(w^\top x) = 0\}.$$

Moreover, the rotation group $SO(m)$ acts on $\mathcal{S}_m^k$ by matrix multiplication, and this action corresponds to applying a rotation to each landmark individually. As $SO(m)$ is compact, this action is proper. However, this action is not free everywhere if $m \geq 3$. This makes the orbit space

$$\Sigma_m^k = \{[x] \mid x \in \mathcal{S}_m^k\} = \mathcal{S}_m^k / SO(m)$$

a "differential manifold with singularities where the action is not free", and these points correspond to matrices of rank $m-2$ or less (i.e. some landmarks are aligned).

By Theorem 4.5 (page 58), the canonical projection $\pi : x \mapsto [x]$ is a submersion. For any $x \in \mathcal{S}_m^k$ and $A \in \mathrm{Skew}(m)$, as for all $t \in \mathbb{R}$, $[\exp(tA)x] = x$, the curve $t \mapsto \exp(tA)x$ is a curve in the fiber through x, so the vertical space at x is

$$V_x = \{Ax \mid A \in \mathrm{Skew}(m)\} = \mathrm{Skew}(m)x.$$

The pullback of Frobenius metric on the pre-shape space allows to define the horizontal spaces:

$$\begin{aligned} H_x &= \{w \in T_x\mathcal{S}_m^k \mid Tr(Axw^\top) = 0 \quad \forall A \in \mathrm{Skew}(m)\} \\ &= \{w \in T_x\mathcal{S}_m^k \mid xw^\top \in \mathrm{Sym}(m)\} \end{aligned}$$

where $\mathrm{Sym}(m)$ is the space of symmetric matrices of size m. Lemma 1 from Nava-Yazdani, Hege, et al. 2020 allows to compute the vertical component of any tangent vector.

Lemma 5.1. *For any $x \in \mathcal{S}_m^k$ and $w \in T_x\mathcal{S}_m^k$, the vertical component of w can be computed as $V_x(w) = Ax$ where A solves the Sylvester equation:*

$$Axx^\top + xx^\top A = wx^\top - xw^\top \tag{5.8}$$

If $\mathrm{rank}(x) \geq m-1$*, the skew-symmetric solution A of equation (5.8) is unique.*

In practice, the Sylvester equation can be solved by an eigenvalue decomposition of $xx^\top$ (see Example 5.2). This defines ver_x, the orthogonal projection on V_x. As $T_x\mathcal{S}_m^k = V_x \oplus H_x$, any tangent vector w at $x \in \mathcal{S}_m^k$ may be decomposed into a horizontal and a vertical component, by solving equation (5.8) to compute $\mathrm{ver}_x(w)$, and then $\mathrm{hor}_x(w) = w - \mathrm{ver}_x(w)$.

As the Frobenius metric is invariant to the action of $SO(m)$, we can define the quotient metric on Σ_m^k, and this makes π a Riemannian submersion. Furthermore, the Riemannian distances d on $\mathcal{S}_m^k$ and d_Σ on Σ_m^k are related by

$$d_\Sigma(\pi(x), \pi(y)) = \inf_{R \in SO(m)} d(x, Ry).$$

The optimal rotation R between any x, y is unique in a subset U of $\mathcal{S}_m^k \times \mathcal{S}_m^k$, which allows to define the align map $\omega : U \to \mathcal{S}_m^k$ that maps (x, y) to Ry. In this case, $d_\Sigma(\pi(x), \pi(y)) = d(x, \omega(x, y))$ and $x\omega(x, y)^\top \in \mathrm{Sym}(m)$. It is useful to notice that $\omega(x, y)$ can be directly computed by a pseudo-singular value decomposition of $xy^\top$(W. S. Kendall and Le 2009).

Remark 5.2. *The alignment problem is similar to the canonical correlation analysis problem (CCA) between two data sets:*

$$\max_{U,V \in O(m)} \langle Ux, Vy \rangle.$$

Finally, in the case of the Kendall shape spaces, the quotient space cannot be seen explicitly as a submanifold of some $\mathbb{R}^N$. Moreover, the projection π and its derivative $d\pi$ cannot be computed. However, the align map amounts to identifying the shape space with a local horizontal section of the pre-shape space, and thanks to the characteristics of Riemannian submersions mentioned in the previous subsections, all the computations can be done in the pre-shape space.

Let Exp, Log, and d denote the operations of the pre-shape space $\mathcal{S}_m^k$, that are given in equation (3.16),(3.17) (page 32). We obtain from Theorem 5.1 for any $x, y \in \mathcal{S}_m^k$ and $v \in T_x\mathcal{S}_m^k$

$$\begin{aligned} \mathrm{Exp}_{\Sigma,[x]}(d\pi_x v) &= \pi(\mathrm{Exp}_x(\mathrm{hor}_x(v))), \\ \mathrm{Log}_{\Sigma,[x]}([y]) &= d\pi_x \mathrm{Log}_x(\omega(x, y)), \\ d_\Sigma([x], [y]) &= d(x, \omega(x, y)). \end{aligned}$$

To end this section, we state Proposition 2 of Kim, I. L. Dryden, et al. 2020, that echoes with Remark 5.1 for the computation of parallel transport.

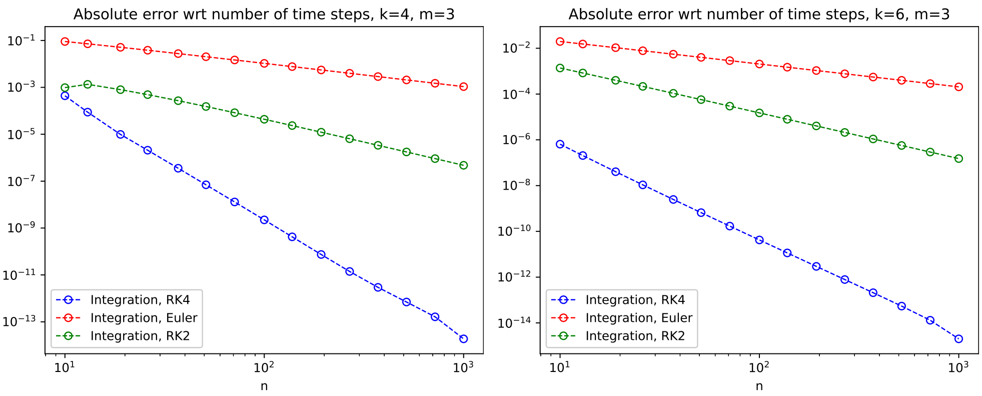


Figure 16: Convergence speed of the integration of equation (5.9) for two orthonormal initial vectors v and $\dot{\gamma}(0)$, with respect to the number of steps. The curve γ is a horizontal geodesic here.

Proposition 5.4 (Kim, I. L. Dryden, et al. 2020)**.** *Let γ be a horizontal C^1-curve in $\mathcal{S}_m^k$ and v be a horizontal tangent vector at $\gamma(0)$. Assume that $\operatorname{rank}(\gamma(s)) \geq m-1$ except for finitely many s. Then the vector field $s \mapsto v(s)$ along γ is horizontal and the projection of $v(s)$ to $T_{[\gamma(s)]}\Sigma_m^k$ is the parallel transport of $d\pi_x v$ along $[\gamma(s)]$ if and only if $s \mapsto v(s)$ is the solution of*

$$\dot{v}(s) = -\operatorname{tr}(\dot{\gamma}(s)v(s)^\top)\gamma(s) + A(s)\gamma(s), \qquad v(0) = v \tag{5.9}$$

where for every s, $A(s) \in \operatorname{Skew}(m)$ is the unique solution to

$$A(s)\gamma(s)\gamma(s)^\top + \gamma(s)\gamma(s)^\top A(s) = \dot{\gamma}(s)v(s)^\top - v(s)\dot{\gamma}(s)^\top. \tag{5.10}$$

Equation (5.9) means that the covariant derivative of $s \mapsto v(s)$ along γ must be a vertical vector at all times, defined by the matrix $A(s) \in \operatorname{Skew}(m)$. These equations can be used to compute parallel transport in the shape space. To compute the parallel transport of $d\pi_x w$ along $[\gamma]$, Kim, I. L. Dryden, et al. propose the following method: one first chooses a discretization time-step $\delta = \frac{1}{n}$, then repeat for every $s = \frac{i}{n}, i = 0 \dots n$

1. Compute $\gamma(s)$ and $\dot{\gamma}(s)$,
2. Solve the Sylvester equation (5.10) to compute $A(s)$ and the r.h.s. of (5.9),
3. Take a discrete Euler step to obtain $\tilde{v}(s+\delta)$
4. Project $\tilde{v}(s+\delta)$ to $T_{\gamma(s)}\mathcal{S}_m^k$ to obtain $\hat{v}(s+\delta)$,
5. Project to the horizontal subspace: $v(s+\delta) \leftarrow \operatorname{hor}(\hat{v}(s+\delta))$
6. $s \leftarrow s+\delta$

We notice that this method can be accelerated by a higher-order integration scheme, such as Runge-Kutta (RK) by directly integrating the system $\dot{v} = f(v,s)$ where f is a smooth map given by equations (5.9) and (5.10). In this case, steps 4. and 5. are not necessary. The precision and complexity of this method is then bound to that of the integration scheme used, as shown in the plot below (Figure 16) for randomly generated orthogonal tangent vectors of unit norm.

5.1.4 Implementation

We now present an abstract class to construct a quotient metric. There are two scenarios. In the first one, a submersion $\pi : E \to M$ is given between the total space and the base manifold, together with a metric on E whose restriction to horizontal spaces is preserved by π. This is the case of the Bures-Wasserstein metric (Examples 5.2, 5.3 and 5.5).

In the second scenario, we are only given

1. a total space E,
2. a group G acting freely, properly and smoothly on E,
3. a metric on E invariant to the action of G.

In this case, the base manifold cannot be represented explicitly, that is, as an embedded submanifold of $\mathbb{R}^N$. Thus, we cannot implement it as an `OpenSet` or `LevelSet` class on its own. However, we can use the properties of the canonical projection, that is a Riemannian submersion, to construct a metric on the quotient space. This metric is in fact defined on horizontal spaces of the total space, and the base manifold is locally identified with a section of the total space. This is the case of Kendall shape spaces (Section 5.1.3).

To model both cases, we use the structure of fiber bundle. The first scenario is a fiber bundle by definition. For the second scenario, we notice that $\pi : E \to M$ and $(G, \cdot)$ form a principal fiber bundle. Conversely, if we endow a principal fiber bundle with a G-invariant metric, we can define a quotient metric on B. We thus choose to implement an abstract `FiberBundle` class, as main ingredient of a `QuotientMetric` class.

As the goal is to compute the Riemannian Exp, Log and distance maps according to equations (5.3), (5.7) and (5.6), we need to specify a section and the corresponding horizontal lift, and the align map. Then the quotient metric is implemented as follows

```python
class QuotientMetric(RiemannianMetric):
    """Quotient metric."""

    def __init__(self, fiber_bundle: FiberBundle, dim: int = None):
        if dim is None:
            if fiber_bundle.base is not None:
                dim = fiber_bundle.base.dim
            elif fiber_bundle.group is not None:
                dim = fiber_bundle.dim - fiber_bundle.group.dim
            else:
                raise ValueError('Either the base manifold, '
                                 'its dimension, or the group acting on the '
                                 'total space must be provided.')
        super().__init__(
            dim=dim, default_point_type=fiber_bundle.default_point_type)

        self.fiber_bundle = fiber_bundle
        self.group = fiber_bundle.group
        self.ambient_metric = fiber_bundle.ambient_metric

    def inner_product(
            self, tangent_vec_a, tangent_vec_b, base_point=None,
            point_above=None):
        """Compute the inner-product of two tangent vectors at a base point."""
        if point_above is None:
            if base_point is not None:
                point_above = self.fiber_bundle.lift(base_point)
            else:
                raise ValueError(
                    'Either a point (of the total space) or a base point (of '
                    'the quotient manifold) must be given.')
        hor_a = self.fiber_bundle.horizontal_lift(tangent_vec_a, point_above)
        hor_b = self.fiber_bundle.horizontal_lift(tangent_vec_b, point_above)
```

```
        return self.ambient_metric.inner_product(hor_a, hor_b, point_above)

    def exp(self, tangent_vec, base_point, **kwargs):
        """Compute the Riemannian exponential of a tangent vector."""
        lift = self.fiber_bundle.lift(base_point)
        horizontal_vec = self.fiber_bundle.horizontal_lift(tangent_vec, lift)
        return self.fiber_bundle.riemannian_submersion(
            self.ambient_metric.exp(horizontal_vec, lift))

    def log(self, point, base_point, **kwargs):
        """Compute the Riemannian logarithm of a point."""
        point_fiber = self.fiber_bundle.lift(point)
        bp_fiber = self.fiber_bundle.lift(base_point)
        aligned = self.fiber_bundle.align(point_fiber, bp_fiber, **kwargs)
        return self.fiber_bundle.tangent_riemannian_submersion(
            self.ambient_metric.log(aligned, bp_fiber), bp_fiber)

    def squared_dist(self, point_a, point_b, **kwargs):
        """Squared geodesic distance between two points."""
        lift_a = self.fiber_bundle.lift(point_a)
        lift_b = self.fiber_bundle.lift(point_b)
        aligned = self.fiber_bundle.align(lift_a, lift_b, **kwargs)
        return self.ambient_metric.squared_dist(aligned, lift_b)
```

Example 5.6: Implementation of the Bures-Wasserstein metric

In the setting of Examples 5.2, 5.3 and 5.5, that corresponds to the first scenario as $E = GL(n)$ and $M = SPD(n)$, we create a new class that corresponds to $GL(n)$ with the fiber bundle structure:

```
class BuresWassersteinBundle(GeneralLinear, FiberBundle):
    def __init__(self, n):
        super().__init__(
            n=n, base=SPDMatrices(n), group=SpecialOrthogonal(n),
            ambient_metric=MatricesMetric(n, n))

    @staticmethod
    def riemannian_submersion(point):
        return Matrices.mul(point, Matrices.transpose(point))

    @staticmethod
    def lift(point):
        return gs.linalg.cholesky(point)

    def tangent_riemannian_submersion(self, tangent_vec, base_point):
        product = Matrices.mul(
            base_point, Matrices.transpose(tangent_vec))
        return 2 * Matrices.to_symmetric(product)

    def horizontal_lift(self, tangent_vec, point_above=None, base_point=None):
        if base_point is None and point_above is not None:
            if point_above is not None:
                base_point = self.riemannian_submersion(point_above)
            else: raise ValueError(
                'Either a point in the fiber or a base point in base manifold)'
                ' must be given.')
        sylvester = gs.linalg.solve_sylvester(
            base_point, base_point, tangent_vec)
        return Matrices.mul(sylvester, point_above)
```

Note that all computations can actually be carried out in closed form (see Bhatia, Jain, et al. 2019), and this specific examples allows to test our general implementation. In this case we used the minimization procedure to compute the align map.

In the second scenario, we only compute in E so both submersion and lift maps are the identity of E. This can be set by default and overridden as in Example 5.6. Furthermore, in that case, tangent vectors to M are identified with horizontal vectors of E, so we need to compute the horizontal decomposition of a tangent vector to E. Thus, either the horizontal or vertical projection needs to be implemented, and the other one can be deduced.

In the first scenario, these are given by using $d\pi$ and the horizontal lift, as vertical vectors are in the kernel of $d\pi$ by definition.

This explains the following `FiberBundle` class (the align method is not duplicated from above in the interest of space):

```
class FiberBundle(Manifold, abc.ABC):
    """Class for (principal) fiber bundles."""

    def __init__(
            self, dim: int, base: Manifold = None,
            group: LieGroup = None, ambient_metric: RiemannianMetric = None,
            group_action=None, **kwargs):

        super().__init__(dim=dim, **kwargs)
        self.base = base
        self.group = group
        self.ambient_metric = ambient_metric

        if group_action is None and group is not None:
            group_action = group.compose
        self.group_action = group_action

    @staticmethod
    def riemannian_submersion(point):
        """Project a point to base manifold."""
        return point

    @staticmethod
    def lift(point):
        """Lift a point to total space."""
        return point

    def align(self, point, base_point,
              max_iter=25, verbose=False, tol=gs.atol):
        """Align point to base_point by optimization in the Lie algebra."""
        pass

    def tangent_riemannian_submersion(self, tangent_vec, base_point):
        """Project a tangent vector to base manifold."""
        return self.horizontal_projection(tangent_vec, base_point)

    def horizontal_lift(self, tangent_vec, point_above=None, base_point=None):
        """Lift a tangent vector to a horizontal vector in the total space."""
        if point_above is None:
            if base_point is not None:
                point_above = self.lift(base_point)
            else:
                raise ValueError(
                    'Either a point (of the total space) or a base point (of'
                    'the base manifold) must be given.')
        return self.horizontal_projection(tangent_vec, point_above)

    def horizontal_projection(self, tangent_vec, base_point):
        """Project to horizontal subspace."""
        try:
            return tangent_vec - self.vertical_projection(
                tangent_vec, base_point)
        except (RecursionError, NotImplementedError):
            return self.horizontal_lift(
```

```python
            self.tangent_riemannian_submersion(
                tangent_vec, base_point), base_point)

    def vertical_projection(self, tangent_vec, base_point, **kwargs):
        """Project to vertical subspace."""
        try:
            return tangent_vec - self.horizontal_projection(
                tangent_vec, base_point)
        except RecursionError:
            raise NotImplementedError
```

Example 5.7: Kendall shape metric

With the above construction of the second scenario, Kendall shape metric is just a subclass of the `QuotientMetric` class. We also add the parallel transport method discussed above and used in Figure 16.

```python
class KendallShapeMetric(QuotientMetric):
    """Quotient metric on the shape space."""

    def __init__(self, k_landmarks, m_ambient):
        bundle = PreShapeSpace(k_landmarks, m_ambient)
        super().__init__(
            fiber_bundle=bundle,
            dim=bundle.dim - int(m_ambient * (m_ambient - 1) / 2))

    def parallel_transport(
            self, tangent_vec_a, tangent_vec_b, base_point, n_steps=100,
            step='rk4'):
        """Compute the parallel transport of a tangent vec along a geodesic."""
        horizontal_a = self.fiber_bundle.horizontal_projection(
            tangent_vec_a, base_point)
        horizontal_b = self.fiber_bundle.horizontal_projection(
            tangent_vec_b, base_point)

        def force(state, time):
            gamma_t = self.ambient_metric.exp(time * horizontal_b, base_point)
            speed = self.ambient_metric.parallel_transport(
                horizontal_b, time * horizontal_b, base_point)
            coef = self.inner_product(speed, state, gamma_t)
            normal = gs.einsum('...,...ij->...ij', coef, gamma_t)

            align = gs.matmul(Matrices.transpose(speed), state)
            right = align - Matrices.transpose(align)
            left = gs.matmul(Matrices.transpose(gamma_t), gamma_t)
            skew_ = gs.linalg.solve_sylvester(left, left, right)
            vertical_ = - gs.matmul(gamma_t, skew_)
            return vertical_ - normal

        flow = integrate(force, horizontal_a, n_steps=n_steps, step=step)
        return flow[-1]
```

Example 5.8: Full rank correlation matrices

As a last example, we implement the metric on the set of Correlation matrices described in Thanwerdas and Xavier Pennec 2021a using our abstract `FiberBundle` and `QuotientMetric` classes.

The set of full-rank correlation matrices $Corr(n)$ is a submanifold of $SPD(n)$ formed by matrices with unit diagonal. Define the action of positive diagonal matrices on $SPD(n)$ by congruence:

$$\triangleright : (D, \Sigma) \in Diag_+(n) \times SPD(n) \mapsto D\Sigma D \in SPD(n).$$

This action is smooth, free, and proper (David and Gu 2019). The quotient manifold $SPD(n)/Diag_+(n)$ is thus well defined, and can be identified with $Corr(n)$ by the map $[\Sigma] \mapsto D_\Sigma^{-1/2} \triangleright \Sigma$ where D_Σ is the diagonal matrix with coefficients Σ_{ii}.

Let $\pi : SPD(n) \to Corr(n)$ be the canonical projection composed with this map, i.e. $\Sigma \mapsto D_\Sigma^{-1/2} \triangleright \Sigma$, and consider the affine-invariant metric defined on $SPD(n)$ by

$$g_\Sigma(V, W) = tr(\Sigma^{-1} V \Sigma^{-1} W).$$

This metric will be detailed in Example 5.14 in Section 5.3. It is clear that it is invariant by the action of diagonal matrices.

Therefore we can define the quotient metric such that π is a Riemannian submersion. The horizontal and vertical spaces are given in Thanwerdas and Xavier Pennec 2021a, Theorem 1. The submersion and its differential, the vertical projection and horizontal lifts are computed in closed form, so we are in the first scenario, and the quotient metric is simply

```python
class FullRankCorrelationAffineQuotientMetric(QuotientMetric):
    """Class for the quotient of the affine-invariant metric."""

    def __init__(self, n):
        super().__init__(
            fiber_bundle=CorrelationMatricesBundle(n=n))
```

where the `FiberBundle` class is used to define `CorrelationMatricesBundle` as follows:

```python
class CorrelationMatricesBundle(SPDMatrices, FiberBundle):
    """Fiber bundle for the quotient metric on correlation matrices."""

    def __init__(self, n):
        super().__init__(
            n=n, base=CorrelationMatrices(n),
            ambient_metric=SPDMetricAffine(n), group_dim=n,
            group_action=CorrelationMatrices.diag_action)

    @staticmethod
    def riemannian_submersion(point):
        """Compute the correlation matrix associated to an SPD matrix."""
        diagonal = Matrices.diagonal(point) ** (-.5)
        aux = gs.einsum('...i,...j->...ij', diagonal, diagonal)
        return point * aux

    def tangent_riemannian_submersion(self, tangent_vec, base_point):
        """Compute the differential of the Riemannian submersion."""
        diagonal_bp = Matrices.diagonal(base_point)
        diagonal_tv = Matrices.diagonal(tangent_vec)

        diagonal = diagonal_tv / diagonal_bp
        aux = base_point * (diagonal[..., None, :] + diagonal[..., :, None])
        mat = tangent_vec - .5 * aux
        return CorrelationMatrices.diag_action(diagonal_bp ** (-.5), mat)

    def vertical_projection(self, tangent_vec, base_point, **kwargs):
        """Compute the vertical projection wrt the affine-invariant metric."""
        n = self.n
        inverse_base_point = GeneralLinear.inverse(base_point)
        operator = gs.eye(n) + base_point * inverse_base_point
        inverse_operator = GeneralLinear.inverse(operator)
        vector = gs.einsum(
            '...ij,...ji->...i', inverse_base_point, tangent_vec)
        diagonal = gs.einsum('...ij,...j->...i', inverse_operator, vector)
        return base_point * (diagonal[..., None, :] + diagonal[..., :, None])

    def horizontal_lift(self, tangent_vec, base_point=None, fiber_point=None):
        """Compute the horizontal lift wrt the affine-invariant metric."""
        if fiber_point is None and base_point is not None:
            return self.horizontal_projection(tangent_vec, base_point)
        diagonal_point = Matrices.diagonal(fiber_point) ** 0.5
        lift = CorrelationMatrices.diag_action(diagonal_point, tangent_vec)
        hor_lift = self.horizontal_projection(lift, base_point=fiber_point)
        return hor_lift
```

5.2 Homogeneous spaces

We now come back to the particular case of homogeneous spaces, that are quotient spaces where the total space is a Lie group $E = G$ under the action of a subgroup $H \subset G$ of G, and $M = G/H$. We focus on Riemannian metrics that are invariant to the group action, that is, for which the action of any $g \in G$ on M is an isometry. This case is fundamental as it is the simplest way in which all the geometry of M is determined by that of G. If additionnally the metric on G is bi-invariant, we say the metric on G/H is normal, and all the computations can be carried in closed-form.

5.2.1 Characterization

Recall that a homogeneous space $M \simeq G/H$ is a manifold with a smooth transitive action of G on M, and corresponds to the orbits of the right-action of a closed subgroup H of G. As the map $\pi : G \mapsto G/H$ is a submersion (see Theorem 4.5 page 58), we shall see that invariant metrics on G/H are in fact particular cases of the quotient metrics of the previous section. Indeed, a particular case of Theorem 5.2 (page 66) gives the following proposition.

Proposition 5.5. *Let G be a Lie group and $H \subsetneq G$ a closed subgroup of G. Write $o = eH = H \in G/H$, and $\mathfrak{g}, \mathfrak{h}$ the Lie algebras of respectively G, H. Let g be a left-invariant Riemannian metric on G, that is also right-invariant by H. Then there exists a unique Riemannian metric on G/H that is invariant to the action of G and such that $d\pi_e$ is an isometry between $\mathfrak{h}^\perp \subset \mathfrak{g}$ and T_oG/H. In fact, π is a Riemannian submersion.*

Indeed, Theorem 5.2 applies with $E = G$ and $G = H$ as the action of H on G is smooth, free and proper (by Theorem 4.6), and the metric on G is invariant to this action. This defines a metric on G/H, and we can show that it is invariant to G. Indeed, for any $x, y \in G$, write $M_{yx^{-1}} : G/H \to G/H$ that maps some $[p]$ to $[yx^{-1}p]$, and $L_{yx^{-1}} : G \to G$ the left translation of G by yx^{-1}. Then by definition of the action of G on M, we have the following commutation

$$\pi \circ L_{yx^{-1}} = M_{yx^{-1}} \circ \pi$$

Differentiating the above expression at x and as $d\pi_x, d\pi_y$ and $d(L_{yx^{-1}})_x$ are isometries, we conclude that $d(M_{yx^{-1}})_{[x]}$ is also an isometry.

5.2.2 Existence

The above proposition thus states that we need a left-invariant metric on G that is also right-invariant by H to define an invariant metric on G/H. There are necessary and sufficient conditions on G and H for these to exist (see e.g. Gallier and Quaintance 2020, Proposition 22 .22).

Proposition 5.6. *If G acts faithfully on G/H and if $\mathfrak{g}$ admits a decomposition $\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{m}$ with $\mathrm{AD}_H(\mathfrak{m}) \subset \mathfrak{m}$, then G-invariant metrics on G/H are in one-to-one correspondence with AD_H-invariant scalar products on $\mathfrak{m}$. These exist if and only if the closure of the group $\mathrm{AD}_H(\mathfrak{m})$ is compact. Conversely, if G/H admits a G-invariant metric, then G admits a left-invariant metric which is right invariant under H, and the restriction of this metric to H is bi-invariant. The decomposition of $\mathfrak{g}$ is given by $\mathfrak{m} = \mathfrak{h}^\perp$ in this case.*

Note that if H is connected, the condition $\mathrm{AD}_H(\mathfrak{m}) \subset \mathfrak{m}$ is equivalent to $[h, \mathfrak{m}] \subset \mathfrak{m}$. If H is compact, then $\mathrm{AD}_H(\mathfrak{m})$ is compact, so a G-invariant metric exists. The above proposition suggests the following definitions due to Nomizu 1954.

Definition 5.7 (Reductive Homogeneous space)**.** Let G be a Lie group and H a closed subgroup of G. Write $\mathfrak{g}, \mathfrak{h}$ their Lie algebra. We say that the homogeneous space G/H is *reductive* if there exists some subspace $\mathfrak{m} \subset \mathfrak{g}$ such that

$$\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{m} \qquad \text{and} \qquad \mathrm{AD}_h(\mathfrak{m}) \subseteq \mathfrak{m} \quad \forall h \in H.$$

In this case, $\mathfrak{m}$ is isomorphic to T_oG/H via $d\pi_e$. In fact, $\mathfrak{h}$ is the vertical subspace at e, and $\mathfrak{m}$ is the horizontal space. The notion of reductive homogeneous space is important because it is a sufficient condition for $M = G/H$ to admit a G-invariant connection, whose geodesics are projections (by π) of the one-parameter subgroups of G. This is not sufficient however to obtain such property from the Levi-Civita connection of some metric, and an additional condition will be given in the next subsection. Accordingly with the decomposition $\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{m}$, any element of the Lie algebra can be decomposed as the sum of an element of $\mathfrak{m}$ and an element of $\mathfrak{h}$. For any $x \in \mathfrak{g}$ we write this decomposition as $x = x_\mathfrak{m} + x_\mathfrak{h}$ where $x_\mathfrak{m} \in \mathfrak{m}$ and $x_\mathfrak{h} \in \mathfrak{h}$.

Example 5.9: Invariant metric on Stiefel manifold

Recall from Example 4.17 (page 60) that with $G = SO(n)$ and

$$H = \left\{ \begin{pmatrix} I_k & 0 \\ 0 & R \end{pmatrix} \mid R \in SO(n-k) \right\} \simeq SO(n-k),$$

we obtain the Stiefel manifold $S(n,k) = G/H$, the set of orthonormal k-frames represented by $n \times k$ orthogonal matrices. The equivalence class of some $Q = (U, U_\perp) \in S(k,n)$ is $[Q] = QH = (U, U_\perp R)$ where R is any matrix in $SO(n-k)$. The canonical projection is therefore given by $\pi : Q = (U, U_\perp) \mapsto U$ and the origin $o = eH$ of G/H is $\pi(I_n) = I_k$. Thus we can write π as the projection on the first k columns. Moreover, $\mathfrak{g}$ is the set of skew-symmetric matrices, so that we have $\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{m}$ with

$$\mathfrak{h} = \left\{ \begin{pmatrix} 0 & 0 \\ 0 & S \end{pmatrix}, \ S \in \mathfrak{so}(n-k) \right\}$$
$$\mathfrak{m} = \left\{ \begin{pmatrix} T & -A^\top \\ A & 0 \end{pmatrix}, \ T \in \mathfrak{so}(k) A \in M_{n-k,k}(\mathbb{R}) \right\}$$

It is straightforward to check that $[\mathfrak{h}, \mathfrak{m}] \subseteq \mathfrak{m}$, and as H is connected, $S(k,n)$ is a reductive homogeneous space.

As H is compact, by Proposition 5.6 there exists a G-invariant metric on $S(k,n)$. Indeed, as the pullback of the Frobenius metric on G is bi-invariant on G, hence on H, Proposition 5.5 applies and the metric on $S(k,n)$ is defined such that $\pi : Q = (U, U_\perp) \mapsto U$ is a Riemannian submersion. For $X = \begin{pmatrix} T & -A^\top \\ A & 0 \end{pmatrix} \in \mathfrak{m}$, $d\pi_o X = \begin{pmatrix} T \\ A \end{pmatrix}$ and we obtain

$$\langle \begin{pmatrix} T \\ A \end{pmatrix}, \begin{pmatrix} S \\ B \end{pmatrix} \rangle \triangleq \frac{1}{2} \operatorname{tr} \left(\begin{pmatrix} T & -A^\top \\ A & 0 \end{pmatrix}^\top \begin{pmatrix} S & -B^\top \\ B & 0 \end{pmatrix} \right) = \frac{1}{2} \operatorname{tr}(T^\top S) + \operatorname{tr}(A^\top B) \tag{5.11}$$

A more convenient way of representing a tangent vector at U is by $X = US + (I - UU^\top)A$. Then one can show that the $SO(n)$-invariant metric defined in equation (5.11) can be written

$$\langle X_1, X_2 \rangle_U = \operatorname{tr}(X_1 (I_k - \frac{1}{2} UU^\top) X_2).$$

For more on this metric, see Gallier and Quaintance 2020, Section 22.5.

5.2.3 Properties

Recall that for a reductive Lie algebra, we have the decomposition: $[X,Y] = [X,Y]_\mathfrak{m} + [X,Y]_\mathfrak{h}$. By using the formulas of the connection of an invariant metric on a Lie group (see Section 4.3) and of a Riemannian submersion (Section 5.1), one can show the following (see e.g. Gallier and Quaintance 2020, Propositions 22.25 and 22.27).

Theorem 5.3. *Let G be a Lie group and H a closed subgroup of G. If there exists an AD_H-invariant inner product on $\mathfrak{m}$, then the Levi-Civita connection of the induced metric on G/H is given by $\nabla_X Y = -\frac{1}{2}[X,Y]_\mathfrak{m}$ and the geodesics are projections of one-parameter subgroups if and only if*

$$\langle x, [z,y]_\mathfrak{m} \rangle + \langle [z,x]_\mathfrak{m}, y \rangle = 0 \quad \forall x, y, z \in \mathfrak{m}. \tag{5.12}$$

In this case, we say that G/H is naturally reductive.

This property allows to derive closed-form expressions for the geodesics in many cases, as one parameter subgroups are given by the matrix exponential in classical Lie groups. A similar behavior holds for curvature and parallel transport. Many formulas implemented in geomstats can be retrieved this way.

Example 5.10: Stiefel Exponential map

Following Example 5.9 and applying theorem 5.3, we can verify that the metric verifies equation (5.12). Therefore, the exponential map is given by

$$\mathrm{Exp}_U\left(\begin{pmatrix} S \\ A \end{pmatrix}\right) = (U, U_\perp) \exp\left(\begin{pmatrix} S & -A^\top \\ A & 0 \end{pmatrix}\right)\begin{pmatrix} I_k \\ 0 \end{pmatrix}$$

This expression can be simplified with a QR decomposition of A, and its inverse (the Logarithm) can be computed recursively see Zimmermann 2017, for more details.

5.3 Symmetric spaces

To conclude this section, we briefly expose symmetric spaces, as they can be defined from a homogeneous space G/H with an additional tool defined on G. This results in one of the most simple geometries, where the geodesics, parallel transport and curvature can be computed in closed form. We start with a more intrinsic definition and will connect with this description after. Similarly to that of homogeneous space, the structure of symmetric space does not necessarily require a Riemannian metric but only an affine connection. We first focus on the Riemannian case for simplicity, and give a few remarks on the more general case, referring to it as the affine symmetric case. This notion was introduced by Cartan 1926 who fully achieved a classification of symmetric spaces. The most complete reference is Helgason 1979, and a good exposition of the non Riemannian case is given in Kobayashi and Nomizu 1996b, Chapter XI and Postnikov 2001, Chapters 4–10.

We first define the geodesic symmetries, the maps defined locally that revert geodesics.

Definition 5.8 (Geodesic symmetry)**.** Let (M, g) be a Riemannian manifold. Let $x \in M$ and $U \subset M$ be an open neighborhood of x such that the exponential map is injective on U. The *geodesic symmetry* at x is the map defined by

$$s_x : \begin{cases} U & \longrightarrow & M \\ y & \longmapsto & \mathrm{Exp}_x(-\mathrm{Log}_x(y)) \end{cases} .$$

It is clear that for any $x \in M$, x is an isolated fixed point of s_x, and that $(ds_x)_x = -\mathrm{id}$, where id is the identity transformation of T_xM.

Definition 5.9 (Locally symmetric space)**.** Let (M, g) be a Riemannian manifold. M is called *locally symmetric* if for any $x \in M$, the geodesic symmetry s_x is an isometry.

Remark 5.3. *This definition is valid in an affine space where the notion of isometry is replaced by the notion of affine map, i.e., that preserve the connection.*

The property that defines locally symmetric spaces has direct consequences on its curvature tensor (Kobayashi and Nomizu 1996b, Chapter XI, Theorem 6.2).

Theorem 5.4. *A Riemannian manifold M is locally symmetric if and only if $\nabla R = 0$.*

We shall say that the curvature of M is *covariantly* constant, and this will simplify the parallel transport on locally symmetric spaces.

Definition 5.10 (Symmetric Space)**.** (M, g) is called (globally) symmetric if the geodesic symmetries are defined on the whole manifold M and are isometries.

The two notions are equivalent up to topological constrains, as stated in the following theorem (Kobayashi and Nomizu 1996b, Chapter XI, Theorems 6.3-6.4).

Theorem 5.5. *A geodesically complete, simply connected, locally symmetric space is globally symmetric. Conversely, every globally symmetric space is geodesically complete.*

We now come to the interesting theorem that relates symmetric spaces to homogeneous spaces (Kobayashi and Nomizu 1996b, Chapter XI, Theorem 6.5).

Theorem 5.6. *The group of isometries of M $\mathrm{Isom}(M)$ is a Lie group that acts transitively on M. M is thus a homogeneous space. Let G be the largest connected group of isometries of M, and consider a reference point $o \in M$. Let H its stabilizer by the action of G. Then H is compact and $M \simeq G/H$.*

Define G as in the above theorem. We can define an additional structure on G that will be the essence that differentiate symmetric spaces from homogeneous spaces.

Remark 5.4.

- *The composition of two geodesic symmetries belongs to G, and is called a transvection.*
- *Let γ be a geodesic through o, then transvections $s_{\gamma(t)} \circ s_o$ are one-parameter subgroups of G.*

Theorem 5.7. *Let $M = G/H$ be a symmetric space, define on G the map $\sigma : g \mapsto s_o \circ g \circ s_o$.*

- *σ is involutive: $\sigma \circ \sigma = Id$,*
- *σ is a group homeomorphism, i.e. $\sigma(g \circ h) = \sigma(g) \circ \sigma(h)$, thus an automorphism,*
- *The set G^σ of fixed point of σ is a closed subgroup, and with G_0^σ its connected component, we have $G_0^\sigma \subset H \subset G^\sigma$ (implying $\dim(H) = \dim(G^\sigma)$).*

Define $\mathfrak{k} = \{v \in \mathfrak{g}, d\sigma_e(v) = v\}$ and $\mathfrak{m} = \{v \in \mathfrak{g}, d\sigma_e(v) = -v\}$. Then $\mathfrak{k}$ coincides with the Lie algebra $\mathfrak{h}$ of H and G/H is naturally reductive with $\mathfrak{g} = \mathfrak{k} \oplus \mathfrak{m}$, i.e. $[\mathfrak{k}, \mathfrak{k}] \subset \mathfrak{k}$ and $[\mathfrak{k}, \mathfrak{m}] \subset \mathfrak{m}$, and we have the additional property:

$$[\mathfrak{m}, \mathfrak{m}] \subset \mathfrak{k}.$$

As a consequence (Theorem 5.3), the geodesics of a symmetric space are thus projections of one parameter subgroups. In fact symmetric spaces are the only homogeneous spaces with an involutive automorphism as described in Theorem 5.7.

Definition 5.11 (Symmetric pair)**.** A *symmetric pair* is a triplet (G, H, σ) where G is a connected Lie group, H a closed subgroup of G and σ is an involutive automorphism of G, such that its set of fixed points G^σ satisfies $G_0^\sigma \subset H \subset G^\sigma$.

Remark 5.5.

- *If there exists $a \in G$ s.t. $a^2 = e$, then $\sigma : g \mapsto a \circ g \circ a^{-1}$ is an involutive automorphism, and its set of fixed points G^σ is a closed (normal) subgroup of G, so that (G, G^σ, σ) forms a symmetric pair.*
- *The inversion $g \mapsto g^{-1}$ is an automorphism if and only if G is commutative. This is very restrictive, and in general the involution is not the inversion map.*

- *In fact one can show (see e.g. Cheeger and Ebin 1975, Proposition 3.37) that a simply connected group G possesses an involutive automorphism σ if and only if its Lie algebra admits a decomposition $\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{m}$ with $[\mathfrak{h}, \mathfrak{h}] \subset \mathfrak{h}$, $[\mathfrak{h}, \mathfrak{m}] \subset \mathfrak{m}$ and $[\mathfrak{m}, \mathfrak{m}] \subset \mathfrak{h}$.*

We now see how to recover a symmetric space from a symmetric pair. Recall that $\pi : g \in G \to gH \in G/H$ is the canonical projection of the quotient G/H, and that G acts on G/H by $g_1 \triangleright (g_2H) = (g_1g_2)H$. Let $o = eH = H$, and $s_o : gH \mapsto \sigma(g)H$, i.e. $s_o \circ \pi = \pi \circ \sigma$.

Theorem 5.8. *Let (G, H, σ) be a symmetric pair such that H and G_0^σ are compact. Define $\mathfrak{h} = \{v \in \mathfrak{g}, d\sigma_e(v) = v\}$ and $\mathfrak{m} = \{v \in \mathfrak{g}, d\sigma_e(v) = -v\}$. Then $M = G/H$ is naturally reductive with the decomposition $g = \mathfrak{h} \oplus \mathfrak{m}$. Furthermore, with the family of symmetries defined at any $x = gH \in M$ by*

$$s_x = g \circ s_o \circ g^{-1},$$

M is a globally symmetric space.

Remark 5.6. *The assumption that H and G_0^σ are compact is sufficient to ensure the existence of a G-invariant metric on G/H such that it is naturally reductive. It is not necessary however to show that G/H is affine symmetric, but the connection may not be the Levi-Civita connection of any metric. If a G-invariant metric exists however on an affine symmetric space $M = G/H$, its Levi-Civita connection coincides with the connection of the affine symmetric structure (Kobayashi and Nomizu 1996b, Chapter XI, Theorem 3.3).*

The last theorem along with Remark 5.5 now make it easier to exemplify the notion of symmetric space.

Example 5.11: Hypersphere as a symmetric space

Recall from Example 4.16 (page 59) that the hypersphere S^d can be seen as the quotient of $G = SO(d+1)$ by $H \simeq SO(d)$ defined by

$$H = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & R \end{pmatrix} \mid R \in SO(d) \right\}.$$

Define $J = \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix}$. Obviously $J^2 = I_{d+1}$ so that the map defined by $\sigma : P \in G \mapsto JPJ \in G$ is an involutive automorphism, and it is straightforward to check that $G^\sigma = H$. Then (G, H, σ) is a symmetric pair and by Theorem 5.8, S^d is a symmetric space. The Lie algebra is decomposed in $\mathfrak{so}(d+1) = \mathfrak{h} \oplus \mathfrak{m}$ with

$$\mathfrak{h} = \left\{ \begin{pmatrix} 0 & 0 \\ 0 & S \end{pmatrix}, \ S \in \mathfrak{so}(d) \right\}$$
$$\mathfrak{m} = \left\{ \begin{pmatrix} 0 & -u^\top \\ u & 0 \end{pmatrix}, \ u \in \mathbb{R}^d \right\}.$$

From the expressions of the Exp and Log map, we can compute the symmetry at $x \in S^d$. For any $y \in S^d$

$$s_x(y) = 2(\langle x, y \rangle)x - y.$$

Similarly, the upper hyperboloid and the Euclidean space R^d are also symmetric. Thus, all constant-curvature spaces are symmetric.

Example 5.12: Grassmann manifold as a symmetric space

The Grassmann manifold $Gr(k,n)$ (Example 4.18 page 60) is the set of k-dimensional subspaces of $\mathbb{R}^n$, and is identified with $O(n)/(O(k)\times O(n-k))$. Define $J = \begin{pmatrix} I_k & 0 \\ 0 & -I_{n-k} \end{pmatrix}$. Obviously $J^2 = I_n$ so that the map defined by $\sigma : P \in G \mapsto JPJ \in G$ is an involutive automorphism with fixed points

$$G^\sigma = \left\{ \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \mid Q \in O(k), R \in O(n-k), \;\; \det(Q)\det(R) = 1 \right\},$$

and $G^\sigma = S(O(k)\times O(n-k))$ with $G_0^\sigma = SO(k)\times SO(n-k)$. Note that if we use $H = G_0^\sigma$, we restrict to *oriented* k-subspaces, and consider in this case the oriented Grassmann manifold. In both cases, we have

$$\mathfrak{h} = \left\{ \begin{pmatrix} S & 0 \\ 0 & T \end{pmatrix}, \; S \in \mathfrak{so}(k), T \in \mathfrak{so}(n-k) \right\}$$
$$\mathfrak{m} = \left\{ \begin{pmatrix} 0 & -A^\top \\ A & 0 \end{pmatrix}, \; A \in M_{n-k,k}(\mathbb{R}) \right\}.$$

Then (G, H, σ) is a symmetric pair and by Theorem 5.8, $Gr(k,n)$ is a symmetric space. Given any $P = QP_kQ^\top \in Gr(k,n)$, the symmetry at P is $s_P : \tilde{P} \mapsto (QJQ^\top)\tilde{P}(QJQ^\top)$. The symmetric space structure allows to deduce many properties of $Gr(k,n)$, namely that it is geodesically complete, hence a complete metric space (by the Hopf-Rinow theorem page 33), and its exponential map is surjective at all points.

Example 5.13: Stiefel manifold as a symmetric space

We now give an example of homogeneous space that is not symmetric: the Stiefel manifold $St(k,n)$. Indeed recall that we have $St(k,n) = G/H$ with $G = SO(n)$ and $H = SO(n-k)$ and reductive decomposition $\mathfrak{so}(n) = \mathfrak{h} \oplus \mathfrak{m}$ with

$$\mathfrak{h} = \left\{ \begin{pmatrix} 0 & 0 \\ 0 & T \end{pmatrix}, \; T \in \mathfrak{so}(n-k) \right\}$$
$$\mathfrak{m} = \left\{ \begin{pmatrix} S & -A^\top \\ A & 0 \end{pmatrix}, \; S \in \mathfrak{so}(k) A \in M_{n-k,k}(\mathbb{R}) \right\}$$

We can check that $[\mathfrak{m}, \mathfrak{m}] \not\subset \mathfrak{h}$, so that $St(k.n)$ is not symmetric with this decomposition.

Example 5.14: The Affine-Invariant metric on SPD matrices

Recall that $SPD(n)$ is a homogenous space with (restricting to a connected component) $G = GL^+(n)$ and $H = SO(n)$ and canonical projection that coincides with $\pi : A \mapsto AA^\top$. Define $\sigma : A \in G \mapsto A^{-\top}$. It is clear that σ is an involutive automorphism with $G^\sigma = H$. Then (G, H, σ) is a symmetric pair and by Theorem 5.8, $SPD(n)$ is a symmetric space. From σ and π we deduce the symmetry for $\Sigma, \Lambda \in SPD(n)$: $s_\Sigma(\Lambda) = \Sigma\Lambda^{-1}\Sigma$.

The affine-invariant (AI) metric is defined as the quotient metric on G/H of the left-invariant metric on G that coincides with the Frobenius metric at I. Its expression is thus at any $\Sigma \in SPD(n)$, for all $V, W \in \mathrm{Sym}(n)$

$$g_\Sigma(V, W) = \mathrm{tr}(\Sigma^{-1}V\Sigma^{-1}W).$$

From the projection of one parameter subgroups we deduce $\forall \Sigma, \Sigma_1, \Sigma_2 \in SPD(n), \forall W \in \mathrm{Sym}(n)$

$$\mathrm{Exp}_\Sigma(W) = \Sigma^{\frac{1}{2}} \exp(\Sigma^{-\frac{1}{2}} W \Sigma^{-\frac{1}{2}}) \Sigma^{\frac{1}{2}},$$
$$\mathrm{Log}_{\Sigma_1}(\Sigma_2) = \Sigma_1^{\frac{1}{2}} \log(\Sigma_1^{-\frac{1}{2}} \Sigma_2 \Sigma_1^{-\frac{1}{2}}) \Sigma_1^{\frac{1}{2}},$$

where when not indexed, exp and log refer to the matrix operators. Finally, let

$$P_t = \Sigma^{\frac{1}{2}} \exp(\frac{t}{2} \Sigma^{-\frac{1}{2}} W \Sigma^{-\frac{1}{2}}) \Sigma^{-\frac{1}{2}}.$$

The parallel transport from Σ along the geodesic with initial velocity $W \in \mathrm{Sym}(n)$ of $V \in \mathrm{Sym}(n)$ a time t is (Yair, Ben-Chen, et al. 2019)

$$\Pi_{0,W}^t V = P_t V P_t^\top.$$

We now focus on the case of Lie groups themselves, that can be seen as symmetric spaces. However, one must be cautious on the structure, either metric or affine that is used. Let G be a connected Lie group. Consider the product group $\tilde{G} = G \times G$ with the involution $\sigma : (g, h) \mapsto (h, g)$. The subgroup of fixed points is the diagonal of $\tilde{G}$, $H = \{(g, g), g \in G\}$, and its Lie algebra is $\mathfrak{h} = \{(x, x), x \in \mathfrak{g}\} \simeq \mathfrak{g}$. We thus see from Proposition 5.6, that for $G = G \times G/G$ to be Riemannian symmetric hence reductive homogeneous, it must admit a bi-invariant metric. There are still three possible reductive decompositions $\mathfrak{m} \oplus \mathfrak{h}$ of the Lie algebra $\tilde{\mathfrak{g}} = \mathfrak{g} \times \mathfrak{g}$ of $\tilde{G}$:

$$\mathfrak{m} = \{(x, 0), x \in \mathfrak{g}\}$$
$$\mathfrak{m} = \{(0, x), x \in \mathfrak{g}\}$$
$$\mathfrak{m} = \{(x, -x), x \in \mathfrak{g}\}$$

With either of these, G is homogeneous reductive, and each decomposition lead to a different connection, called respectively *left, right or mean* connection. These are known as the Cartan-Schouten connections (Lorenzi and Xavier Pennec 2011). On can show that with the last connection, G is an affine symmetric space with the symmetries $\forall g \in G,\ s_g : h \mapsto gh^{-1}g$ (Xavier Pennec, Sommer, et al. 2020, Theorem 5.8). If G admits a bi-invariant metric, this connection is the Levi-Civita connection of the bi-invariant metric and G is a Riemannian symmetric space. An example of this case is the group of rotation matrices $SO(n)$. On the contrary, $SE(n)$ does not admit any bi-invariant metric, so it does not admit a Riemannian symmetric structure that coincides with its Cartan-Schouten connection.

The properties of symmetric spaces allow to compute the geodesics and parallel transport in closed-form, by the projection of the one-parameter subgroups of the Lie group G. We rarely use explicitly this structure in geomstats, as all the results were derived and implemented case by case. The structure is however useful to compute the curvature. Indeed, using the identification of $\mathfrak{m}$ with $T_o(M)$ induced by the restriction of $d\pi$ to $\mathfrak{m}$, we have for all $X, Y, Z \in \mathfrak{m}$

$$R(X, Y)Z = -[[X, Y], Z]$$

Moreover under topological conditions (M is simply connected and irreducible), O'Neill's formula (5.4) (page 65) simplifies and the sectional curvature is for $u, v \in \mathfrak{m}$ orthogonal

$$\kappa(u, v) = \frac{1}{2}\langle [[u, v], v], u\rangle - \frac{1}{2}\langle [[v, u], u], v\rangle.$$

To conclude this section, symmetric spaces offer a useful framework for statistics on manifolds beyond the convenience of the closed form solutions for the geodesics and parallel transport. Firstly, the normal distribution can be defined on all symmetric spaces, by generalizing the property that its maximum likelihood estimate coincides with the least-square problem of the Frechet mean (Said, Hajri, et al. 2018). The normalizing factor (or partition function) does not depend on the mean and can be computed in closed form thanks a decomposition of the Lie algebra.

Moreover, many computations such as interpolation or sampling can be performed in the subspace $\mathfrak{m}$ of the Lie algebra and be projected back to the symmetric space as in Gawlik and Leok 2018; Munthe-Kaas, Quispel, et al. 2014; Barp, Kennedy, et al. 2019. Finally, there is a large theory of harmonic analysis on symmetric spaces (see Terras 1988), and limit theorems for stochastic processes such as Brownian motion.

In this section, we focused on the common implementation of Riemannian metrics that arise from group actions. The first case is for invariant metrics on Lie groups themselves, for which the invariance allows to reformulate all the geometric operations as algebraic operations in the Lie algebra. In particular, we applied this reasoning to derive a parallel transport equation, that lead to a stable implementation in geomstats.

The second case is that of quotient metrics, that arise on the orbits of the group actions. We identified the key ingredients to a common implementation in geomstats, and focused on the Kendall shape spaces and Bures-Wasserstein metric to exemplify these. This formulation allowed to easily implement parallel transport on Kendall shape spaces, and is to the best of our knowledge the only open-source Python implementation of these spaces. The space of correlation matrices seen as the quotient of SPD manifolds by the action of diagonal matrices is also implemented in geomstats, and others spaces such as the space of positive semi-definite matrices will be implemented in the near future.

Two particular instances of quotient spaces were then introduced: reductive homogeneous spaces and symmetric spaces. They are in fact direct generalizations of Lie groups and many operations from quotient spaces simplify. These concepts are however not used in the present implementation, as their properties actually allows to derive more efficient closed-form solutions.

6 Statistics and machine learning with Geomstats

To conclude this section, we demonstrate the use of geomstats to perform statistics on manifold-valued data. The strength of the package is that learning algorithms are defined as external estimators that take the geometric object as input. This happens thanks to the standardized interface of the classes, and ensures that all learning tools be available for all the manifolds and metrics. The obtained flexibility allows to compare the impact of the different metrics on the learning results. We give a brief introduction to geometric statistics, and the interested reader is referred to Xavier Pennec, Sommer, et al. 2020, Chapter 2 for the theoretical exposition. An introductory paper gathering more examples using geomstats was presented at the Scipy Conference 2020 in Austin, Texas (Nina Miolane, Guigui, et al. 2020). More generally, statistical methods for objects living in stratified and infinite dimensional spaces that may not be manifolds are developed in the emerging domain of Object-Oriented Data Analysis (OODA) Marron and I. Dryden 2021.

6.1 Probability distributions and sampling

Given a probability measure, one can define random variables valued in a manifold M as follows.

Definition 6.1. Let $(\Omega, \mathcal{F}, \Pr)$ be a probability space, where Ω is a sample space, $\mathcal{F}$ a σ-algebra* and Pr a probability measure*. Then a *random variable* in the Riemannian

*Defined in Appendix A.

manifold M is an $\mathcal{F}$-measurable function $X : \Omega \to M$.

Recall that a probability measure is an assignment of a size to each subset (e.g. length, area or volume in Euclidean spaces) such that the size of the entire space is $\Pr(\Omega) = 1$. A Riemannian metric offers a convenient framework to define probability distributions on manifolds, as it defines a volume measure, that in turn, allows to define the probability measure. This measure is used as reference-measure of probability densities.

Consider a Euclidean space $\mathbb{R}^n$, and an orthonormal basis $(e_1, \ldots, e_n)$ of $\mathbb{R}^n$. Let $(a_1, \ldots, a_m)$ be a set of vectors, and A the matrix whose columns are formed by the coordinates of the a_i. Then the volume spanned by $(a_1, \ldots, a_m)$ is given by $\det(A) = \sqrt{\det G}$ where $G = A^\top A$, i.e. $A_{ij} = \langle a_i, a_j \rangle$ (see Figure 17.)

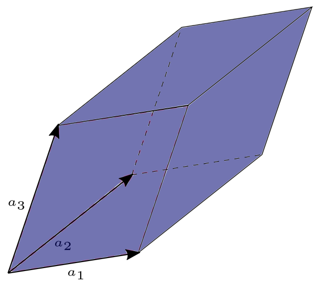


Figure 17: Volume spanned by three vectors in $\mathbb{R}^3$.

Now in a Riemannian manifold (M, g), for any $x \in M$, one may choose an orthonormal basis $(a_1, \ldots a_d)$ of $T_x M$ with respect to g. g is then represented by the SPD matrix $G_{ij} = \langle a_i, a_j \rangle$, so that the volume spanned by these vectors is $\sqrt{\det G}$.

Definition 6.2 (Riemannian volume form)**.** An oriented Riemannian manifold (M, g) has a natural *volume form* defined by $d\,\mathrm{Vol}(x) = \sqrt{\det(g(x))}dx$.

The volume form provides the way to define integrals on M, i.e. it defines a measure, so that some probability distributions can be expressed by their densities with respect to that measure.

Definition 6.3. The random variable X has *density* f if $\forall \mathcal{X} \in \mathcal{B}(M)$, $\Pr(X \in \mathcal{X}) = \int_{\mathcal{X}} f(y)dv(y)$ and $\Pr(M) = 1$.

Example 6.1: Uniform distribution

Let M be a compact Riemannian manifold, for example the hypersphere, the special orthogonal group of the Grassmann manifold. Then a uniform distribution on M has density

$$f(x) = \frac{1}{\mathrm{Vol}(M)}.$$

Example 6.2: Gaussian distribution

Let M be a Riemannian Symmetric space. A Gaussian distribution with mean and concentration (μ, Γ) is defined by the density

$$f(x) = \alpha(\Gamma, \mu) \exp(-\log_\mu(x)^\top \Gamma \log_\mu(x)).$$

It is the entropy maximizing distribution (Xavier Pennec 2006), and in the isotropic, the maximum likelihood estimator of μ coincides with the Frechet Mean (Xavier Pennec, Sommer, et al. 2020, Section 2.5.1). The normalizing constant is explicitely computed in negative curvature spaces (Said, Hajri, et al.

2018).

Sampling A common task in statistics is to draw samples from a given probability distribution on a manifold. This might be motivated by inference tasks where the posterior distributions has constrained parameters, in testing goodness of fit for exponential families, or to generate samples of data to test our learning algorithms (Diaconis, Holmes, et al. 2013; Barp, Kennedy, et al. 2019, and references therein).

However this is does not reduce to sampling from usual distributions even when a parametrization of the manifold is available. Indeed, the curvature of the space generally deforms or stretches the densities. This is illustrated in the following examples.

Example 6.3: Uniform distribution on the sphere

Consider the unit sphere $S^2 \subset \mathbb{R}^3$, with the spherical coordinates from the north pole $e_0 = (0, 0, 1)$: $x = \sin(\phi)\cos(\theta), y = \sin(\phi)\sin(\theta), z = \cos(\phi)$. The volume element (here the area) is $d\,\mathrm{Vol}(\theta, \phi) = \sin(\phi) d\theta d\phi$. A naive attempt at sampling from the uniform distribution on the sphere would be to sample θ uniformly in $[0, 2\pi)$ and ϕ uniformly in $[0, \pi)$.

However, near $\phi = 0$ and $\phi = \pi$, as $\sin(\phi)$ there is less density than with respect to the usual measure of the 2d-plane. We thus expect points to accumulate in a non-uniform way around the poles of the sphere.

In fact, if Y be a Gaussian vector in $\mathbb{R}^d$ with mean 0 and covariance identity. Then $X = \frac{Y}{\|Y\|}$ is uniformly distributed in S^d. The two sampling strategies are tested with geomstats and shown on Figure 18.

```
space = Hypersphere(2)
n_samples = 5000
uniform_param = gs.random.rand(
    n_samples, 2) * gs.pi * gs.array([1., 2.])[None, :]
naive_samples = space.spherical_to_extrinsic(uniform_param)
uniform_samples = space.random_uniform(n_samples)
```

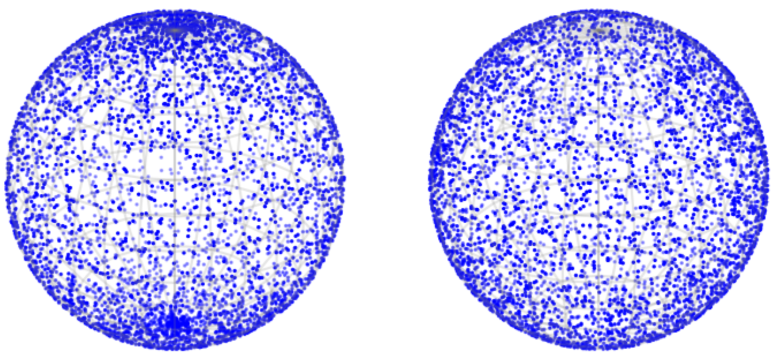


Figure 18: Comparison of the naive sampling from a uniform distribution on the spherical coordinates (left) and a more appropriate scheme that respects the anisotropy of the scheme (right). In the naive case, points accumulate near the poles and are sparser around the equator.

Example 6.4: Uniform distribution on the Special Orthogonal group

Let $SO(n)$ be the group of unit determinant orthogonal matrices of size $n \times n$. Let Y be a matrix of size $n \times n$ with independent standard normal distribution on each coefficient. The QR decomposition of Y defines $Y = XR$ where $X \in SO(n)$ and R upper triangular. Then X is uniformly distributed on $SO(n)$ w.r.t. to the Haar measure, which coincides with the Riemannian measure (Eaton 1983).

Example 6.5: Uniform distribution on Stiefel manifold

Recall the Stiefel manifold is the set of orthonormal k-frames in $\mathbb{R}^n$. It can be represented by the set of matrices $\{U \in \mathbb{R}^{n \times k} | U^\top U = I_k\}$. Let Z be a matrix with i.i.d standard normal distribution on the entries. Then $X = Z(Z^\top Z)^{-1/2}$ is uniformly distributed (Chikuse 2003, Theorem 2.2.1). Note that X is a factor of the polar decomposition of Z, representing the *orientation* of Z.

Example 6.6: Uniform distribution on Grassmann manifold

The Grassmann manifold is the set of k-dim subspaces of $\mathbb{R}^n$. It can be represented by the the set of projection matrices of size $n \times n$, i.e. symmetric rank-k P such that $P^2 = P$. Let Z be a matrix with i.i.d standard normal distribution on the entries. Then $X = Z(Z^\top Z)^{-1/2}Z$ is uniformly distributed (Chikuse 2003, Theorem 2.2.2). Note that $X = YY^\top$ for Y uniformly distributed on the Stiefel manifold.

These examples are particular cases where simple recipes were available to sample from the uniform distribution. They are implemented in geomstats. Sampling from non-uniform distribution is usually intractable and we resort to simulation methods from the literature. In particular, in geomstats we focused on rejection sampling, that consists in sampling from a proposal distribution whose density is greater than the target density, and accepting the samples with probability corresponding to the ratio of the two densities[2]. We do not detail this procedure in this manuscript but refer to Wood 1994; Hauberg 2018 for examples of rejection sampling algorithms for non-uniform distributions on the hypersphere.

6.2 Distance-based algorithms

A large class of machine learning algorithms only require computing a matrix where each entry is the distance between a pair of samples. This is the case for examples of the nearest neighbors methods or hierarchical clustering. These can readily be used by computing the distance matrix with a geomstats metric. We give below a toy example on the sphere, with data generated with two von Mises distributions (Example 6.7).

Example 6.7: Hierarchical clustering on the sphere

The coaches of the French skiing Olympic team studies the solid angles of inclination of their skier during a slalom. To first filter left and right turns, they use a hierarchical clustering algorithm on the sphere of solid angles.

```
space = Hypersphere(2)
metric = space.metric
```

[2] see the Wikipedia page for more details.

To test the method, they first generate a toy dataset with von Mises-Fisher distributions

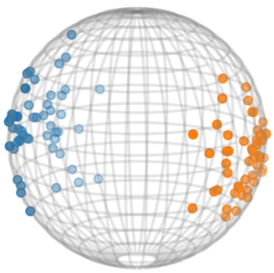
```
n_clusters = 2
n_samples = 50

left = sphere.random_von_mises_fisher(kappa=10, n_samples=n_samples)
right = -sphere.random_von_mises_fisher(kappa=10, n_samples=n_samples)
dataset = gs.concatenate((left, right), axis=0)
```

A geomstats model with the Riemannian distance can then be used to compute the distances. We use `n_clusters=None, distance_threshold=0` in order to compute the full tree and plot a dendrogram (as in the scikit-learn example).

```
model = AgglomerativeHierarchicalClustering(
    n_clusters=None, distance=metric.dist, distance_threshold=0)
model.fit(dataset)
plot_dendrogram(model)
```

The cluster assignments can also be computed by setting `n_cluster=2`.

```
model = AgglomerativeHierarchicalClustering(n_clusters, distance=metric.dist)
model.fit(dataset)
clustering_labels = model.labels_
```



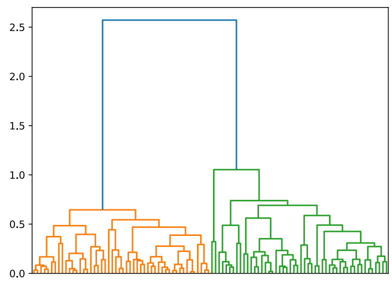


Figure 19: Toy experiment of a hierarchical clustering experiment. The data is sampled from two von Mises-Fisher distributions (left). A dendrogram is computed from the distance matrix (right).

6.3 The Fréchet mean

Let (M, g) be a Riemannian manifold and let $x_1, \ldots, x_n \in M$ be independent identically distributed (i.i.d.) sample data points. The Euclidean sample mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is not defined unless we consider $M \subset \mathbb{R}^N$, but in this case $\bar{x}$ may not lie on M.

Definition, Existence, Uniqueness In fact, the mean is characterized by the property that it minimises the sum of squared distances to the data points. This property was used by Fréchet 1948 to generalize the notion of mean to metric spaces, and later by Hermann Karcher in Riemannian manifolds. For a historical note and corresponding references, we refer to Karcher 2014.

Definition 6.4 (Fréchet mean)**.** Let (M, g) be a Riemannian manifold with Riemannian distance function d and let $x_1, \ldots, x_n \in M$ be an i.i.d. data set. The sample *Fréchet* *mean* is defined as the set of minimizers of the sum-of-squared distances, i.e.,

$$\bar{x} = \arg\min_{x \in M} \sum_{i=1}^{n} d(x_i, x)^2.$$

With some nuance in the requirements on the minimum (local or global), the Fréchet mean is also referred to as Karcher mean or Riemannian barycenter, or even Wasserstein barycenter in optimal transport. Note that as it is defined by an optimization, the Fréchet mean may not exist, or not be unique, depending on the properties of the distance function. For instance, the completeness of the metric is a sufficient condition to guarantee the existence of the Fréchet mean of a finite set of points (Xavier Pennec, Sommer, et al. 2020, Chapter 2).

Theorem 6.1. *Let M be a complete metric space. Then the Fréchet mean of any finite set of points $x_1, \dots, x_n$ exists.*

On the other hand, uniqueness depends on the convexity of the distance function and can be related to the sign of the curvature (Karcher 1977; W. S. Kendall 1990; Afsari 2011).

Theorem 6.2. *Let M be a complete Riemannian manifold with sectional curvature bounded above by some δ, and let $\operatorname{inj}(M)$ be its injectivity radius. If samples $x_1, \dots, x_n \in M$ are contained in a geodesic ball of radius*

$$r = \frac{1}{2} \min \left(\operatorname{inj}(M), \frac{\pi}{\sqrt{\delta}} \right)$$

then the Fréchet mean $\bar{x}$ is unique.

Note that in the above, $\frac{\pi}{\sqrt{\delta}}$ is interpreted as ∞ if $\delta \leq 0$. This means that in complete manifolds of non-positive curvature, the Fréchet mean is always uniquely defined, as in the Euclidean case. On the other hand, this is not the case in spaces of positive curvature. For example, on the sphere, Theorem 6.2 ensures that the mean is unique if all the data points lie on the same open hemisphere, but the simple data set constituted by two antipodal points ($\{x, -x\}$ for any $x \in S^d$) exemplifies non-uniqueness of the mean: a great circle in this case.

The asymptotic properties of the Fréchet mean were studied in Bhattacharya and Patrangenaru 2005, where a law of large numbers and a central limit theorem are established, and prove the relevance of this notion of mean.

Characterization and Estimation Suppose that the data points $x_1, \dots, x_n$ lie in a ball B of radius $r < \frac{\operatorname{inj}(M)}{2}$ as in Theorem 6.2. Then the squared distance is convex and is obtained by measuring the length of a minimizing geodesic and can be written $d^2(x, y) = \| \operatorname{Log}_x(y) \|_x^2$ for any $x, y \in B$. One can show that the gradient of $y \mapsto d^2(x, y)$ is $-2 \operatorname{Log}_y(x) \in T_y M$ (see the first variation formula in Lafontaine, Gallot, et al. 2004, Theorem 3.31).

Remark 6.1. *Recall that the gradient of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as the adjoint to its differential, i.e.*

$$\forall (x, v) \in TM, \quad \langle \operatorname{grad}_x(f), v \rangle \triangleq df_x v.$$

This definition extends to functions defined on Riemannian manifolds, but the gradient thus depends on the metric, and is sometimes referred to as the Riemannian gradient. It is also named natural gradient in information geometry, in the case of manifolds of parameters of families of distributions with the Fisher-Rao metric. If the manifold is embedded in some $\mathbb{R}^N$ and equipped with the pullback metric, it can be obtained by projecting the usual gradient to the tangent space at the point where the function is being differentiated, similarly to the Levi-Civita connection.

By definition, the Fréchet mean must be a critical point of the sum-of-squared distances function, so that it verifies

$$\sum_{i=1}^{n} \mathrm{Log}_{\bar{x}}(y) = 0. \tag{6.1}$$

A simple strategy to estimate the sample Fréchet mean is thus to use a fixed-point iterative algorithm until equation (6.1) is verified. An update at iteration t is performed along a geodesic: $x^{t+1} = \mathrm{Exp}_{x^t}(\gamma \sum_{i=1}^{n} \mathrm{Log}_{x^t}(y))$, where γ is a step size. Higher-order methods use adaptive step size, approximates of the Hessian function or its link with curvature to improve the performance of the estimation. Some of these are implemented in the `FrechetMean` module of geomstats.

Example 6.8: Fréchet mean on the sphere

The following is an illustrative example with simulated data. A scientific expedition of 15 boats is collecting samples from the Pacific ocean to measure the quantity of micro-plastics in the water. Every month, they gather all the samples in one boat that goes back to their home harbour for the analyses. As they have been wandering around the Pacific for some time, we sample their positions from a spherical normal distribution around their last meeting point.

```
space = Hypersphere(2)
n_samples = 15
last_meeting_point = gs.array([0., 1., 0.])
samples = space.random_riemannian_normal(
    mean=last_meeting_point, precision=10, n_samples=15)
```

The geometers onboard compute the Fréchet mean of their positions to choose the meeting spot that minimises the sum-of-squared distances to this gathering (supposing no wind or currents).

```
estimator = FrechetMean(space.metric)
estimator.fit(samples)
new_meeting_point = estimator.estimate_
```

Their position (black) and the next meeting point (red) are displayed in Figure 20. The trajectories they have to follow are drawn in green, and are geodesics.

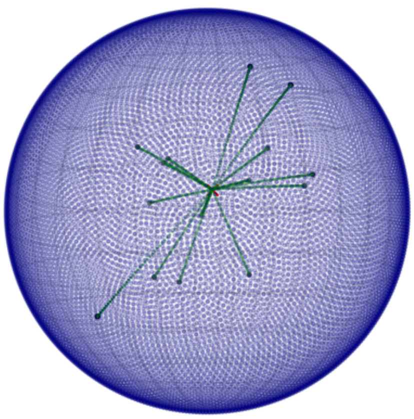


Figure 20: Fréchet mean on the sphere

The Fréchet mean $\bar{x}$ is key to define many other statistical and learning tools, e.g.

the sample covariance matrix:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Log}_{\bar{x}} x_i \otimes \mathrm{Log}_{\bar{x}} x_i. \tag{6.2}$$

One can then parametrize densities from their mean and covariance, as in Chevallier and Guigui 2020. The proposed model allows to define a simple density estimation scheme.

One can also generalize the K-means clustering algorithm and the minimum distance to mean (MDM) classification algorithm with Fréchet means. These are implemented in the classes `RiemannianKMeans` and `RiemannianMinimumDistanceToMeanClassifier` in geomstats. Moreover, one can use the Fréchet mean to linearize the data by lifting it to the tangent space at the Fréchet mean, i.e. to consider the transformed data set $(\tilde{x}_1, \dots, \tilde{x}_n)$ where

$$\tilde{x}_i = \mathrm{Log}_{\bar{x}}(x_i).$$

This is equivalent to using a Taylor expansion at the first order of the Riemannian exponential map around 0: $x_i = \mathrm{Exp}_{\bar{x}}(\tilde{x}_i) = \bar{x} + \tilde{x}_i + O(t^2)$. As $T_{\bar{x}}M$ is a vector space, all the usual statistical and machine learning tools can be used off-the-shelf on the transformed data set. This is implemented as a `Transformer` from the scikit-learn package in the `ToTangent` class of geomstats, and can be used in scikit-learn's pipeline.

6.4 Generalizations of PCA

The other fundamental tools to analyse data in vector spaces are the sample covariance matrix, and Principal Component Analysis (PCA). The aim of PCA is to find the sequence of subspaces such that data projected on these subspaces have maximum variance, or equivalently minimum reconstruction error (i.e. sum of squared distance to the original points). This equivalence is no longer true in Riemannian manifolds, as Pythagorean theorem is not true.

We describe the principal geodesic analysis (PGA), that aims at minimizing the reconstruction error when projecting data on a geodesic submanifold (Fletcher, Lu, et al. 2004). In the simplest forward fashion, the mean is first computed, and the first component is a geodesic from the mean. This is close to a slightly different procedure called Geodesic PCA (Huckemann, Hotz, et al. 2010). The projection of a data point x on a geodesic γ with initial velocity v at the mean $\bar{x}$ is defined by

$$\pi_{\bar{x},v}(x) = \arg\min_{t\in\mathbb{R}} d^2\left(x, \mathrm{Exp}_{\bar{x}}(tv)\right).$$

There is no guaranty that this projection exists, but if the data is not too spread out, one can hope that there exists a convex neighborhood where the exponential map is injective that contains x and a portion of the geodesic. The projection is solved by a gradient descent, as in the case of the Fréchet mean, where the gradient of the exponential is computed by automatic differentiation.

The next step is to minimize the overall reconstruction error, given a dataset $x_1, \dots, x_n$, we look for the initial velocity of a geodesic as:

$$v^* = \arg\min_{v\in T_{\bar{x}}M} \sum_{i=1}^{n} d^2\left(x_i, \pi_{\bar{x},v}(x_i)\right). \tag{6.3}$$

This time computing the gradient of the objective function seems more complicated, as n minimization problems are already solved by a gradient descent to evaluate the function. However Ablin, Peyré, et al. 2020 show that the gradient of a function defined by a minimum could be computed efficiently by automatic differentiation of the gradient descent approximation. We exemplify this in the example of the sphere. Of course this example is particularly simple as both Exp and Log maps are computed with closed form solutions whose derivatives are known, and some work remains to use automatic differentiation when the Log map itself is hard to compute.

Example 6.9: Principal geodesic analysis on earth

The following is an illustrative example with simulated data. The boats of the scientific exploration of Example 6.8 are in fact too busy collecting data and cannot make route to the Fréchet mean. Thankfully, the geometers onboard are expert users of geomstats and compute a PGA of their positions, so that one boat can visit the rest of the fleet.

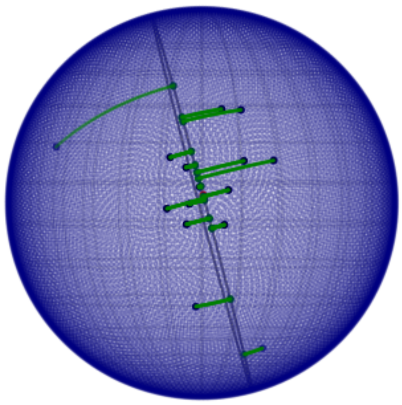


Figure 21: Example of PGA on the sphere. The target data are the blue dots, the projected values are in green, the first geodesic subspace is in black, and the mean in red.

A general simplification if the data is not too far from the mean is to approximate the projection operator by the Log map: $\text{Log}_{\bar{x}}(\pi_{\bar{x},v}(x_i)) \approx \text{Log}_{\bar{x}}(x_i)$. With this approximation, PGA resumes to computing the logarithms of all the data points and computing a usual PCA of the obtained tangent vectors. This procedure is more generally called *tangent PCA* and is implemented in the `TangentPCA` class of geomstats.

6.5 Geodesic Regression

Similarly to the mean, the linear regression can be generalized to *geodesic regression* on Riemannian manifolds by solving a least-square fitting problem. Given target points $y_1, \dots, y_n \in M$, and data $t_1, \dots, t_n \in \mathbb{R}$, we seek the geodesic that best approximates the data:

$$\min_{(p,v)\in TM} \sum_{i=1}^{n} d^2 \left(\text{Exp}_p(t_i v), y_i\right), \tag{6.4}$$

where d and Exp are the Riemannian distance and exponential maps. When the metric is Euclidean, this coincides with the usual linear regression problem. However, there is no closed-form solution in general and the problem must be solved by optimization.

To differentiate the objective function of equation (6.4), one needs to compose the gradients of the squared distance with that of the exponential map. The gradient of the squared distance is proportional to the Riemannian logarithm, as noted above. On the other hand, the gradient of the Exp map is usually computed via Jacobi fields. However, to avoid implementing those in geomstats we chose to leverage automatic differentiation tools to compute the extrinsic gradient, and to project it to the right tangent space, as explained in Remark 6.1.

To solve the optimization problem, either a Riemannian gradient descent, or an extrinsic one with `scipy`'s solver is used.

Example 6.10: Geodesic Regression above earth

The following is an illustrative example with simulated data. The European Space Agency sends a mission to run some maintenance on their geo-stationary satellites. As these are all at the same altitude, they lie on a sphere, and the geometers seek an optimal trajectory.

```
space = Hypersphere(2)
metric = space.metric
```

The maintenance should be done as close as possible to the end of the fuel tanks, which are known from previous missions. One vessel will leave earth, reach the first point of the trajectory called point γ, and split in two parts that will each go in opposite directions given by $\beta \in T_\gamma S^2$.

We use data from the previous mission to generate random satellite positions:

```
n_samples = 50
data = gs.random.rand(n_samples)
data -= gs.mean(data)

previous_gamma = space.random_uniform()
beta = space.to_tangent(5. * gs.random.rand(3), previous_gamma)
target = metric.exp(data[:, None] * beta, previous_gamma)
```

And add some noise because satellites did not stay on the previous misison's trajectory

```
normal_noise = gs.random.normal(size=(n_samples, 3))
noise = space.to_tangent(normal_noise, target) / gs.pi / 2
target = metric.exp(noise, target)
```

The optimal trajectory is computed by fitting a geodesic regression model with the Riemannian gradient descent:

```
gr = GeodesicRegression(space, algorithm='riemannian')
gr.fit(data, target, compute_training_score=True)
gamma_new, beta_new = gr.intercept_, gr.coef_
```

We can measure the mean squared error (MSE) with respect to the previous γ and β, and the determination coefficient from the noise level. We obtain MSEs around 10^{-3} and $R^2 = 0.96$.

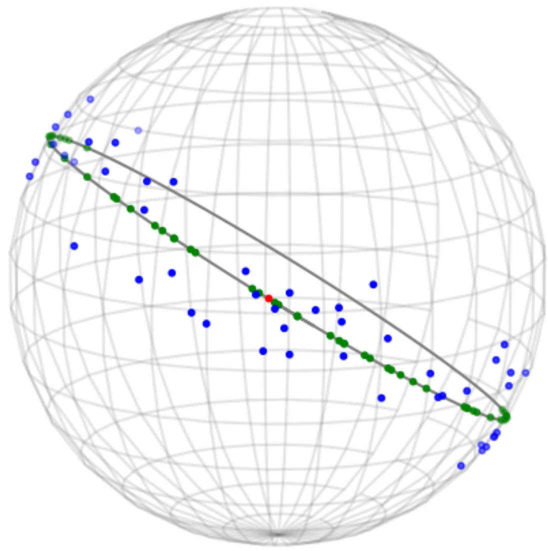


Figure 22: Example of geodesic regression on the sphere. The target data are the blue dots, the fitted values are in green, the regression geodesic in black, and the intercept (or γ in this example) in red.

7 Conclusion

In this paper, we introduced notions of differential geometry that form the essential building blocks of geometric statistics. Our developments target an audience with a general background in mathematics. We focused on embedded manifolds of $\mathbb{R}^N$, and highlighted the different ways of defining a manifold and a Riemannian metric. We detailed the notions of curvature, geodesic, distance and parallel transport. We also presented the concepts of Lie groups, group actions and quotient space, with an emphasis on Riemannian metrics that are invariant to a group action.

These differential geometric notions are key to the architecture of the geomstats library. They drove our recent contributions to the package, that aimed at making it more faithful to mathematical theory, more robust and more modular. We exemplified the geometric concepts with the most common manifolds encountered in mathematical text books as well as in applications. Our examples include code snippets using geomstats to demonstrate how to leverage differential geometry in practical use cases.

Lastly, we gave an introduction to geometric statistic tools such as the Fréchet mean, principal geodesic analysis and geodesic regression. We illustrated these on toy examples using synthetic datasets on manifolds. These geometric statistical tools are implemented in the geomstats package with a common high-level interface, following the Scikit-Learn syntax. Consequently, geometric statistics become available to any data scientist on a wide variety of practical problems.

We hope that the concepts presented here will drive scientists to use, and contribute to, geometric statistics with the geomstats library. Future developments of the geomstats package will integrate additional statistical methodologies published by the geometric statistics community, such as wrapped Gaussian processes, multivariate and polynomial geodesic regression. We will also extend the scope of geomstats and include computational methods for information geometry, a field at the intersection of geometry and statistics closely related to geometric statistics. Another module on stratified spaces such as graph and tree spaces is currently being developed, and raises fundamental methodological questions as these spaces are not smooth manifolds but unions of smooth manifolds equipped with a distance function. Together, these advances aim to provide mathematically-grounded foundations for computational geometric statistics.

Acknowledgment This work was funded by the ERC grant Nr. 786854 G-Statistics from the European Research Council under the European Union's Horizon 2020 research and innovation program. It was also supported by the French government through the 3IA Côte d'Azur Investments ANR-19-P3IA-0002 managed by the National Research Agency. The authors are grateful to Adele Myers for her corrections and suggestions to improve this manuscript.

A Lexicon

Definition A.1 (Topology)**.** Let M be a set and $\mathcal{P}(M)$ the set of subsets of M. Then a *topology* on M is a set $\Theta \in \mathcal{P}(M)$ such that

- $\emptyset \in \Theta$ and $M \in \Theta$
- $\forall U, V \in \Theta, U \cap V \in \Theta$
- $C \subseteq \Theta \implies \bigcup_{U \in C} U \in \Theta$

The sets in Θ are called *open* sets, and a set S is said to be *closed* if and only if $M \backslash S \in \Theta$. Such a pair (M, Θ) is called a topological space.

Definition A.2 (Hausdorff)**.** A topological space (M, Θ) is said to be *Hausdorff* if, for any two distinct points $p, q \in M$, there exist open neighborhoods of p and q with empty intersection.

Definition A.3. A topological space (M, Θ) is second-countable if there exists some countable collection $\mathcal{A} = \{U_i\}_{i \in \mathbb{N}}$ of open sets of M such that any open set can be written as a union of elements of $\mathcal{A}$

Definition A.4 (Connected)**.** A topological space (M, Θ) is said to be *connected* unless there exist two non empty open sets $A, B \in \Theta$ such that $A \cap B \neq 0$ and $M = A \cup B$.

Equivalently, M is connected if and only if the only subsets that are both open and closed are M itself and the empty set $\emptyset$.

Definition A.5 (Group)**.** A *group* is a couple $(G, \cdot)$ where G is a nonempty set, and $\cdot : G \times G \to G$ is a map such that

- $\exists e \in G, \forall g \in G,\ e \cdot g = g \cdot e = g,$
- $\forall g \in G, \exists h,\ g \cdot h = h \cdot g = e.$ We define the inverse of g for $\cdot$ as $g^{-1} = h$ in this case.

Definition A.6 (Homomorphism)**.** Les $(G, \cdot)$ and $(H, \bullet)$ be two groups, and $f : G \to H$ be a map. Then f is a *homomorphism* if for any $x, y \in G$, $f(x \cdot y) = f(x) \bullet f(y)$.

Definition A.7 (Algebra)**.** An *algebra* over a field K is a vector space $(A, +, \cdot)$ over K equipped with a bilinear multiplicative law $\otimes : A \times A \to A$ such that

- (distributivity) $\forall x, y, z \in A$, $(x+y) \otimes z = x \otimes z + y \otimes z$ and $z \otimes (x+y) = z \otimes x + z \otimes y$
- (compatibility with scalars) $\forall a, b \in K, \forall x, y \in A$, $(ax) \otimes (by) = (ab)x \otimes y$.

Definition A.8 (Injective-Surjective-Bijective map)**.** Let E, F be two sets and $f : E \to F$ a map between E and F. Then we say that

- f is *injective* if for every $x, x' \in E$, $x \neq x' \implies f(x) \neq f(x')$,
- f is *surjective* if for every $y \in F$, there exists $x \in E$ such that $y = f(x)$,
- f is *bijective* if it is both injective and surjective.

Definition A.9 (Continuous map)**.** Let E, F be two topological spaces. $f : E \to F$ is *continuous* of C^0 if for every open set $U \subset F$, its preimage $f^{-1}(U)$ by f is an open set of E.

Definition A.10 (Homeomorphism)**.** Let $f : E \to F$ be a map between two topological spaces. f is called a *homeomorphism* if it has the following properties:

- f is a bijection,
- f is continuous,
- the inverse f^{-1} of f is continuous.

Definition A.11 (Differential map)**.** Let $p, n \in \mathbb{N}$, $f : U \subset \mathbb{R}^n \to \mathbb{R}^p$ a map defined on an open set U, and $x_0 \in U$. We say that f is *differentiable* at x_0 if there exists a linear map L defined in $\mathbb{R}^n$ such that

$$\forall h, \quad f(x_0 + h) = f(x_0) + L(h) + o(\|h\|).$$

In that case, L is unique and is called the differential of f at x_0, and written df_{x_0}.

Definition A.12 (Class C^k)**.** Let $p, n \in \mathbb{N}$, $f : U \subset \mathbb{R}^n \to R^p$ a map defined on an open set U. f is C^1 if it is differentiable on U and the map $x \mapsto df_x$ is continuous on U. Similarly we say that f is C^k or of class C^k for $k \in \mathbb{N} \cup \{\infty\}$ if f is k-times differentiable.

Definition A.13 (C^k-diffeomorphism)**.** Let $k \in (\mathbb{N} \setminus \{0\}) \cup \{\infty\}$ and let $f : U \to V$ be a map between two open sets of $\mathbb{R}^n$. f is called a *diffeomorphism* of class C^k if it has the following properties:

- f is a bijection,
- f is of class C^k,
- the inverse f^{-1} of f is C^k.

Definition A.14 (Inner product)**.** Let E be a real vector space. An inner product is a symmetric positive-definite bilinear map $\langle \cdot, \cdot \rangle : E \times E \to \mathbb{R}$, i.e. $\forall x, y \in E, \langle x, y \rangle = \langle y, x \rangle$ and $\langle x, x \langle > 0 \iff x \neq 0$.

Definition A.15 (σ-algebra)**.** Let M be a set and $\mathcal{P}(M)$ the set of subsets of M. A subset $\Sigma \subseteq \mathcal{P}(M)$ is called *σ-algebra* if it has the three following properties:

- It is closed under complementation: for any set $S \in \Sigma$, $M \setminus S \in \Sigma$;
- M is an element of Σ: $M \in \Sigma$;
- Σ is closed under countable unions: $\forall (S_i)_{i \in \mathbb{N}}$, $\bigcup_i S_i \in \Sigma$.

Definition A.16 (Borel σ-algebra)**.** Let M be a topological set. The *Borel* σ-algebra $\mathcal{B}(M)$ on M is the smallest σ-algebra that contains all the open sets of M.

Definition A.17 (Probability measure)**.** Let $\mathcal{F}$ be a σ-algebra over a set Ω. A probability measure Pr is a function $\Pr : \mathcal{F} \to [0, 1]$ such that:

- Pr is σ-additive: $\forall (S_i)_{i \in \mathbb{N}}$, $\Pr(\bigcup_i S_i) \in \Sigma_i \Pr(S_i)$.
- Pr has unit mass: $\Pr(\Omega) = 1$

B The special Euclidean group with an anisotropic metric

B.1 Geodesics

```
import matplotlib.pyplot as plt

import geomstats.backend as gs
import geomstats.visualization as visualization
from geomstats.algebra_utils import from_vector_to_diagonal_matrix
from geomstats.geometry.invariant_metric import InvariantMetric
from geomstats.geometry.special_euclidean import SpecialEuclidean

SE2_GROUP = SpecialEuclidean(n=2, point_type='matrix')
N_STEPS = 15


def main():
    """Plot geodesics on SE(2) with different structures."""
    theta = gs.pi / 4
    initial_tangent_vec = gs.array([
        [0., - theta, 1],
        [theta, 0., 1],
        [0., 0., 0.]])
    t = gs.linspace(0, 1., N_STEPS + 1)
    tangent_vec = gs.einsum('t,ij->tij', t, initial_tangent_vec)

    fig = plt.figure(figsize=(10, 10))
    maxs_x = []
    mins_y = []
    maxs = []
    for i, beta in enumerate([1., 2., 3., 5.]):
        ax = plt.subplot(2, 2, i + 1)
        metric_mat = from_vector_to_diagonal_matrix(gs.array([1, beta, 1.]))
        metric = InvariantMetric(SE2_GROUP, metric_mat, point_type='matrix')
        points = metric.exp(tangent_vec, base_point=SE2_GROUP.identity)
        ax = visualization.plot(
            points, ax=ax, space='SE2_GROUP', color='black',
            label=r'$\beta={}$'.format(beta))
        mins_y.append(min(points[:, 1, 2]))
        maxs.append(max(points[:, 1, 2]))
        maxs_x.append(max(points[:, 0, 2]))
        plt.legend(loc='best')

    for ax in fig.axes:
        x_lim_inf, _ = plt.xlim()
        x_lims = [x_lim_inf, 1.1 * max(maxs_x)]
        y_lims = [min(mins_y) - .1, max(maxs) + .1]
        ax.set_ylim(y_lims)
        ax.set_xlim(x_lims)
        ax.set_aspect('equal')
    plt.savefig('../figures/geo-se2.eps', bbox_inches='tight', pad_inches=0)
    plt.show()


if __name__ == '__main__':
    main()
```

B.2 Curvature

This metric is used already in Example 4.13. From the structure constants and equation (4.4), we can compute the associated Christoffel symbols at identity for the frame

$(\tilde{e_1}, \dots \tilde{e_6})$. Let $\tau = (\sqrt{\beta} + \frac{1}{\sqrt{\beta}})$. We obtain

$$\Gamma_{ij}^k = \frac{1}{2\sqrt{2}} \ \text{ if } \ ijk \ \text{ is a cycle of } [1,2,3], \tag{B.1}$$

$$\Gamma_{15}^6 = -\Gamma_{16}^5 = -\frac{2}{\tau}\Gamma_{24}^6 = \frac{2}{\tau}\Gamma_{26}^4 = \frac{2}{\tau}\Gamma_{34}^5 = -\frac{2}{\tau}\Gamma_{35}^4 = \frac{1}{\sqrt{2}}, \tag{B.2}$$

and all the others are null.

Lemma B.1. *$(SE(3), g)$ is locally symmetric, i.e. $\nabla R = 0$, if and only if $\beta = 1$.*

We can now prove lemma B.1, formulated as: $(SE(3), g)$ is locally symmetric, i.e. $\nabla R = 0$, if and only if $\beta = 1$. This is valid for any dimension $d \geq 2$ provided that the metric matrix G is diagonal, of size $d(d+1)/2$, with ones everywhere except one coefficient of the translation part.

Proof. For $\beta = 1$, $(SE(d), g)$ is isometric to $(SO(d) \times \mathbb{R}^d, g_{rot} \oplus g_{trans})$. As the product of two symmetric spaces is again symmetric, $(SE(d), g)$ is symmetric.

We prove the contraposition of the necessary condition. Let $\beta \neq 1$. We give i, j, k, l s.t. $(\nabla_{e_i} R)(e_j, e_k)e_l \neq 0$:

$$\begin{aligned}
(\nabla_{e_3} R)(e_3, e_2)e_4 &= \nabla_{e_3}(R(e_3, e_2)e_4) - R(e_3, \nabla_{e_3} e_2)e_4 - R(e_3, e_2)\nabla_{e_3} e_4 \\
&= \nabla_{e_3}(R(e_3, e_2)e_4) + \frac{1}{\sqrt{2}} R(e_3, e_1)e_4 - \frac{\tau}{2\sqrt{2}} R(e_3, e_2)e_5.
\end{aligned}$$

And from the above

$$\begin{aligned}
R(e_3, e_2)e_4 &= \nabla_{e_3}\nabla_{e_2} e_4 - \nabla_{e_2}\nabla_{e_3} e_4 - \nabla_{[e_3, e_2]} e_4 \\
&= -\frac{\tau}{2\sqrt{2}} \nabla_{e_3} e_6 - \nabla_{e_2} e_5 + \frac{1}{\sqrt{2}} \nabla_{e_1} e_4 \\
&= 0. \\
R(e_3, e_1)e_4 &= \nabla_{e_3}\nabla_{e_1} e_4 - \nabla_{e_1}\nabla_{e_3} e_4 - \nabla_{[e_3, e_1]} e_4 \\
&= -\frac{\tau}{2\sqrt{2}} \nabla_{e_1} e_5 - \frac{1}{\sqrt{2}} \nabla_{e_2} e_4 \\
&= -\frac{\tau}{4} e_6 + \frac{\tau}{4} e_6 = 0. \\
R(e_3, e_2)e_5 &= \nabla_{e_3}\nabla_{e_2} e_5 - \nabla_{e_2}\nabla_{e_3} e_5 - \nabla_{[e_3, e_2]} e_5 \\
&= \frac{\tau}{2\sqrt{2}} \nabla_{e_2} e_4 + \frac{1}{\sqrt{2}} \nabla_{e_1} e_5 \\
&= -\frac{\tau^2}{8} e_6 + \frac{1}{2} e_6 = \frac{1}{2}(1 - \frac{\tau^2}{4}) e_6.
\end{aligned}$$

And therefore

$$\beta \neq 1 \implies \tau = (\sqrt{\beta} + \frac{1}{\sqrt{\beta}}) \neq 2 \implies (\nabla_{e_3} R)(e_3, e_1)e_4 = -\frac{\tau}{4\sqrt{2}}(1 - \frac{\tau^2}{4}) e_6 \neq 0,$$

which proves Lemma B.1. □

B.3 One parameter subgroups

Note that these don't depend on the choice of the metric.

```python
import matplotlib.pyplot as plt

import geomstats.backend as gs
import geomstats.visualization as visualization
from geomstats.geometry.special_euclidean import SpecialEuclidean

SE2_GROUP = SpecialEuclidean(n=2, point_type='matrix')
N_STEPS = 30
end_time = 2.7

theta = gs.pi / 3
initial_tangent_vecs = gs.array([
    [[0., - theta, 2], [theta, 0., 2], [0., 0., 0.]],
    [[0., - theta, 1.2], [theta, 0., 1.2], [0., 0., 0.]],
    [[0., - theta, 1.6], [theta, 0., 1.6], [0., 0., 0.]]])
t = gs.linspace(-end_time, end_time, N_STEPS + 1)

fig = plt.figure(figsize=(6, 6))
for tv, col in zip(initial_tangent_vecs, ['black', 'y', 'g']):
    tangent_vec = gs.einsum('t,ij->tij', t, tv)
    group_geo_points = SE2_GROUP.exp(tangent_vec)
    ax = visualization.plot(
        group_geo_points, space='SE2_GROUP', color=col)
ax = visualization.plot(
    gs.eye(3)[None, :, :], space='SE2_GROUP', color='slategray')
ax.set_aspect('equal')
ax.axis("off")
plt.savefig('../figures/exponential_se2.eps')
plt.show()
```

References

Ablin, Pierre, Gabriel Peyré, and Thomas Moreau (July 2020). “Super-Efficiency of Automatic Differentiation for Functions Defined as a Minimum”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 32–41.

Absil, Pierre-Antoine, Robert Mahony, and Rodolphe Sepulchre (2010). “Optimization On Manifolds: Methods and Applications”. In: *Recent Advances in Optimization and Its Applications in Engineering*. Ed. by Moritz Diehl, Francois Glineur, Elias Jarlebring, and Wim Michiels. Berlin, Heidelberg: Springer, pp. 125–144. ISBN: 978-3-642-12597-3. DOI: 10.1007/978-3-642-12598-0_12.

Absil, Pierre-Antoine and Jérôme Malick (Jan. 2012). “Projection-like Retractions on Matrix Manifolds”. In: *SIAM Journal on Optimization* 22.1, pp. 135–158. DOI: 10.1137/100802529.

Afsari, Bijan (2011). “Riemannian Lp Center of Mass: Existence, Uniqueness, and Convexity”. In: *Proceedings of the American Mathematical Society* 139.2, pp. 655–673. ISSN: 0002-9939.

Arnold, Vladimir (1966). “Sur la géométrie différentielle des groupes de Lie de dimension infinie et ses applications à l’hydrodynamique des fluides parfaits”. In: *Annales de l’institut Fourier* 16.1, pp. 319–361. ISSN: 0373-0956. DOI: 10.5802/aif.233.

Atkin, Christopher J. (1975). “The Hopf-Rinow Theorem Is False in Infinite Dimensions”. In: *Bulletin of the London Mathematical Society* 7.3, pp. 261–266. ISSN: 1469-2120. DOI: 10.1112/blms/7.3.261.

Balzin, Edouard (n.d.). “Lectures for MAA 306 Course “Topics in Differential Geometry””. In: (), p. 102.

Barachant, Alexandre (2015). *PyRiemann: Python Package for Covariance Matrices Manipulation and Biosignal Classification with Application in Brain Computer Interface*.

Barczyk, Martin, Silvère Bonnabel, Jean-Emmanuel Deschaud, and Francois Goulette (Nov. 2015). “Invariant EKF Design for Scan Matching-Aided Localization”. In: *IEEE Transactions on Control Systems Technology* 23.6, pp. 2440–2448. ISSN: 1558-0865. DOI: 10.1109/TCST.2015.2413933.

Barp, Alessandro, Anthony Kennedy, and Mark Girolami (Mar. 2019). “Hamiltonian Monte Carlo on Symmetric and Homogeneous Spaces via Symplectic Reduction”. In: arXiv: 1903.02699.

Becigneul, Gary and Octavian Ganea (May 2019). “Riemannian Adaptive Optimization Methods”. In: *7th International Conference on Learning Representations (ICLR)*.

Bendokat, Thomas, Ralf Zimmermann, and Pierre-Antoine Absil (Nov. 2020). *A Grassmann Manifold Handbook: Basic Geometry and Computational Aspects*. Technical Report UCL-INMA-2020.07. arXiv: 2011.13699.

Berger, Marcel (2003). *A Panoramic View of Riemannian Geometry*. Berlin, Heidelberg: Springer. ISBN: 978-3-540-65317-2. DOI: 10.1007/978-3-642-18245-7.

Bhatia, Rajendra, Tanvi Jain, and Yongdo Lim (2019). “On the Bures–Wasserstein Distance between Positive Definite Matrices”. In: *Expositiones Mathematicae* 37.2, pp. 165–191. DOI: 10.1016/j.exmath.2018.01.002.

Bhattacharya, Rabi and Vic Patrangenaru (June 2005). “Large Sample Theory of Intrinsic and Extrinsic Sample Means on Manifolds—II”. In: *The Annals of Statistics* 33.3, pp. 1225–1259. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/009053605000000093.

Boumal, Nicolas (Apr. 2022). *An Introduction to Optimization on Smooth Manifolds*.

Cartan, Elie (1926). “Sur une classe remarquable d’espaces de Riemann”. In: *Bulletin de la Société Mathématique de France* 54, pp. 214–264. DOI: 10.24033/bsmf.1105.

— (1930). *La théorie des groupes finis et continus et l’analysis situs*. Vol. 42. Mémorial des sciences mathématiques. Gauthier-Villars et cie.

Cendra, Hernán, Darryl D. Holm, Jerrold E. Marsden, and Tudor S. Ratiu (1998). "Lagrangian Reduction, the Euler-Poincaré Equations, and Semidirect Products". In: *American Mathematical Society Translations* 186.1, pp. 1–25. ISSN: 0065-9290.

Censi, Andrea (2012). *PyGeometry: Library for Handling Various Differentiable Manifolds.*

Cheeger, Jeff and David Ebin (1975). *Comparison Theorems in Riemannian Geometry.* Vol. 9. North-holland Amsterdam.

Chevallier, Emmanuel and Nicolas Guigui (2020). "A Bi-Invariant Statistical Model Parametrized by Mean and Covariance on Rigid Motions". In: *Entropy* 22.4, p. 432. DOI: 10.3390/e22040432.

Chikuse, Yasuko (2003). *Statistics on Special Manifolds.* Lecture Notes in Statistics. New York: Springer-Verlag. ISBN: 978-0-387-00160-9. DOI: 10.1007/978-0-387-21540-2.

Cohen, Taco S, Mario Geiger, and Maurice Weiler (2019). "A General Theory of Equivariant CNNs on Homogeneous Spaces". In: *Advances in Neural Information Processing Systems.* Vol. 32. Curran Associates, Inc.

David, Paul and Weiqing Gu (2019). "A Riemannian Structure for Correlation Matrices". In: *Operators and Matrices* 13.3, pp. 607–627.

Diaconis, Persi, Susan Holmes, and Mehrdad Shahshahani (Jan. 2013). "Sampling from a Manifold". In: *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, pp. 102–125. DOI: 10.1214/12-IMSCOLL1006.

Dryden, Ian L., Alexey Koloydenko, and Diwei Zhou (Sept. 2009). "Non-Euclidean Statistics for Covariance Matrices, with Applications to Diffusion Tensor Imaging". In: *Annals of Applied Statistics* 3.3, pp. 1102–1123. ISSN: 1932-6157, 1941-7330. DOI: 10.1214/09-AOAS249.

Dryden, Ian L. and Kanti V. Mardia (Sept. 2016). *Statistical Shape Analysis: With Applications in R.* John Wiley & Sons. ISBN: 978-0-470-69962-1.

Eaton, Morris L. (1983). *Multivariate Statistics: A Vector Space Approach.* Probability and Statistics Series. Wiley. ISBN: 978-0-471-02776-8.

Fey, Matthias and Jan E. Lenssen (Apr. 2019). "Fast Graph Representation Learning with PyTorch Geometric". In: arXiv: 1903.02428.

Fletcher, Thomas, Conglin Lu, Stephen M Pizer, and Sarang Joshi (2004). "Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape". In: *IEEE transactions on medical imaging* 23.8, pp. 995–1005.

Fréchet, Maurice (1948). "Les éléments aléatoires de nature quelconque dans un espace distancié". In: *Annales de l'institut Henri Poincaré* 10.4, pp. 215–310.

Gallier, Jean and Jocelyn Quaintance (2020). *Differential Geometry and Lie Groups: A Computational Perspective.* Geometry and Computing. Springer International Publishing. ISBN: 978-3-030-46039-6. DOI: 10.1007/978-3-030-46040-2.

Gawlik, Evan S. and Melvin Leok (June 2018). "Interpolation on Symmetric Spaces Via the Generalized Polar Decomposition". In: *Foundations of Computational Mathematics* 18.3, pp. 757–788. ISSN: 1615-3375, 1615-3383. DOI: 10.1007/s10208-017-9353-0.

Gay-Balmaz, François, Darryl D. Holm, David M. Meier, Tudor S. Ratiu, and François-Xavier Vialard (Aug. 2012). "Invariant Higher-Order Variational Problems II". In: *Journal of Nonlinear Science* 22.4, pp. 553–597. ISSN: 0938-8974, 1432-1467. DOI: 10.1007/s00332-012-9137-2. arXiv: 1112.6380.

Guigui, Nicolas, Elodie Maignant, Alain Trouvé, and Xavier Pennec (July 2021). "Parallel Transport on Kendall Shape Spaces". In: *GSI 2021 - 5th Conference on Geometric Science of Information.* Vol. 12829. Lecture Notes in Computer Science. Springer, Cham, pp. 103–110. DOI: 10.1007/978-3-030-80209-7_12.

Guigui, Nicolas and Xavier Pennec (July 2021). "A Reduced Parallel Transport Equation on Lie Groups with a Left-Invariant Metric". In: *GSI 2021 - 5th Conference on Geometric Science of Information.* Vol. 12829. Lecture Notes in Computer Science. Springer, Cham, pp. 119–126. DOI: 10.1007/978-3-030-80209-7_14.

Hauberg, Søren (July 2018). "Directional Statistics with the Spherical Normal Distribution". In: *2018 21st International Conference on Information Fusion (FUSION)*, pp. 704–711. DOI: 10.23919/ICIF.2018.8455242.
— (Sept. 2019). "Only Bayes Should Learn a Manifold (on the Estimation of Differential Geometric Structure from Data)". In: arXiv: 1806.04994.
Helgason, Sigurdur (Feb. 1979). *Differential Geometry, Lie Groups, and Symmetric Spaces*. Academic Press. ISBN: 978-0-08-087396-1.
Huckemann, Stephan, Thomas Hotz, and Axel Munk (2010). "Intrinsic Shape Analysis: Geodesic PCA for Riemannian Manifolds modulo Isometric Lie Group Actions". In: *Statistica Sinica* 20.1, pp. 1–58. ISSN: 10170405.
Karcher, Hermann (1977). "Riemannian Center of Mass and Mollifier Smoothing". In: *Communications on Pure and Applied Mathematics* 30.5, pp. 509–541. ISSN: 1097-0312. DOI: 10.1002/cpa.3160300502.
— (July 2014). "Riemannian Center of Mass and so Called Karcher Mean". In: arXiv: 1407.2087.
Kendall, Wilfrid S. (1990). "Probability, Convexity, and Harmonic Maps with Small Image I: Uniqueness and Fine Existence". In: *Proceedings of the London Mathematical Society* s3-61.2, pp. 371–406. ISSN: 1460-244X. DOI: 10.1112/plms/s3-61.2.371.
Kendall, Wilfrid S. and Huiling Le (2009). "Statistical Shape Theory". In: *New Perspectives in Stochastic Geometry*, pp. 348–373.
Kim, Kwang-Rae, Ian L. Dryden, Huiling Le, and Katie E. Severn (Feb. 2020). "Smoothing Splines on Riemannian Manifolds, with Applications to 3D Shape Space". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. ISSN: 1467-9868.
Kobayashi, Shoshichi and Katsumi Nomizu (Feb. 1996a). *Foundations of Differential Geometry, Volume 1*. Wiley. ISBN: 978-0-471-15733-5.
— (Feb. 1996b). *Foundations of Differential Geometry, Volume 2*. Wiley. ISBN: 978-0-471-15732-8.
Kochurov, Maxim (2019). *Geoopt: Riemannian Adaptive Optimization Methods with Pytorch Optim*.
Kolev, Boris (Jan. 2004). "Lie Groups and Mechanics: An Introduction". In: *Journal of Nonlinear Mathematical Physics* 11.4, pp. 480–498. ISSN: 1402-9251, 1776-0852. DOI: 10.2991/jnmp.2004.11.4.5. arXiv: math-ph/0402052.
Kühnel, Line and Stefan Sommer (Sept. 2017). "Computational Anatomy in Theano". In: *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*, pp. 164–176. DOI: 10.1007/978-3-319-67675-3_15.
Kühnel, Line, Stefan Sommer, and Alexis Arnaudon (Sept. 2019). "Differential Geometry and Stochastic Dynamics with Deep Learning Numerics". In: *Applied Mathematics and Computation* 356, pp. 411–437. ISSN: 0096-3003. DOI: 10.1016/j.amc.2019.03.044.
Lafontaine, Jacques, Sylvestre Gallot, and Dominique Hulin (2004). *Riemannian Geometry*. Springer Verlag.
Le, Huiling and David G. Kendall (1993). "The Riemannian Structure of Euclidean Shape Spaces: A Novel Environment for Statistics". In: *The Annals of Statistics* 21.3, pp. 1225–1271. ISSN: 0090-5364.
Lee, John M. (2003). *Introduction to Smooth Manifolds*. Springer Science & Business Media. ISBN: 978-0-387-95448-6.
— (2018). *Introduction to Riemannian Manifolds*. Vol. 176. Graduate Texts in Mathematics. Cham: Springer International Publishing. ISBN: 978-3-319-91754-2. DOI: 10.1007/978-3-319-91755-9.
Lorenzi, Marco and Xavier Pennec (Sept. 2011). "Geodesics, Parallel Transport & One-parameter Subgroups for Diffeomorphic Image Registration". In: *Proceedings of the 3rd Workshop on Math Fund of Comp Anat*. Ed. by Xavier Pennec, Sarang Joshi, Mads Nielsen, Sarang, Nielsen, and Mads. Toronto, Canada, pp. 64–74.

Louis, Maxime, Raphaël Couronné, Igor Koval, Benjamin Charlier, and Stanley Durrleman (June 2019). "Riemannian Geometry Learning for Disease Progression Modelling". In: *Information Processing in Medical Imaging*. Springer, Cham, pp. 542–553. DOI: 10.1007/978-3-030-20351-1_42.

Mardia, Kanti V. and Peter E. Jupp (Sept. 2009). *Directional Statistics*. John Wiley & Sons. ISBN: 978-0-470-31781-5.

Marle, Charles-Michel (May 2005). "The Works of Charles Ehresmann on Connections: From Cartan Connections to Connections on Fibre Bundles". In: *Geometry and Topology of Manifolds*. Vol. 76. Bedlewo, Poland, pp. 65–86.

Marron, J.S. and I.L. Dryden (2021). *Object Oriented Data Analysis*. Chapman & Hall/CRC monographs on statistics & applied probability. CRC Press. ISBN: 9781032114804. URL: https://books.google.fr/books?id=X2yjzgEACAAJ.

Marsden, Jerrold E. and Tudor S. Ratiu (2009). "Mechanical Systems: Symmetries and Reduction". In: *Encyclopedia of Complexity and Systems Science*. Ed. by Robert A. Meyers. New York, NY: Springer, pp. 5482–5510. ISBN: 978-0-387-30440-3. DOI: 10.1007/978-0-387-30440-3_326.

McAlpin, John Harris (1965). *Infinite Dimensional Manifolds and Morse Theory*. Columbia University.

Meghwanshi, Mayank, Pratik Jawanpuria, Anoop Kunchukuttan, Hiroyuki Kasai, and Bamdev Mishra (2018). *McTorch, a Manifold Optimization Library for Deep Learning*.

Milnor, John (Sept. 1976). "Curvatures of Left Invariant Metrics on Lie Groups". In: *Advances in Mathematics* 21.3, pp. 293–329. ISSN: 00018708. DOI: 10.1016/S0001-8708(76)80002-3.

— (1984). "Remarks on Infinite-Dimensional Lie Groups". In: *Relativity, groups and topology. 2*.

Miolane, N. and X. Pennec (2015). "Computing Bi-Invariant Pseudo-Metrics on Lie Groups for Consistent Statistics". In: *Entropy* 17.4. ISSN: 10994300. DOI: 10.3390/e17041850.

Miolane, Nina, Nicolas Guigui, Hadi Zaatiti, Christian Shewmake, Hatem Hajri, et al. (July 2020). "Introduction to Geometric Learning in Python with Geomstats". In: *SciPy 2020 - 19th Python in Science Conference*, p. 48. DOI: 10.25080/Majora-342d178e-007.

Munthe-Kaas, H. Z., G. R. W. Quispel, and A. Zanna (Mar. 2014). "Symmetric Spaces and Lie Triple Systems in Numerical Analysis of Differential Equations". In: *BIT Numerical Mathematics* 54.1, pp. 257–282. ISSN: 1572-9125. DOI: 10.1007/s10543-014-0473-5.

Myers, Sumner B. and Norman E. Steenrod (1939). "The Group of Isometries of a Riemannian Manifold". In: *Annals of Mathematics* 40.2, pp. 400–416. ISSN: 0003-486X. DOI: 10.2307/1968928.

Nava-Yazdani, Esfandiar, Hans-Christian Hege, Tim J. Sulllivan, and Christoph von Tycowicz (May 2020). "Geodesic Analysis in Kendall's Shape Space with Epidemiological Applications". In: *Journal of Mathematical Imaging and Vision* 62.4, pp. 549–559. ISSN: 1573-7683. DOI: 10.1007/s10851-020-00945-w.

Nickel, Maximillian and Douwe Kiela (2017). "Poincaré Embeddings for Learning Hierarchical Representations". In: *Advances in Neural Information Processing Systems 30*. Ed. by I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, et al. Curran Associates, Inc., pp. 6338–6347.

Nomizu, Katsumi (Jan. 1954). "Invariant Affine Connections on Homogeneous Spaces". In: *American Journal of Mathematics* 76.1, p. 33. ISSN: 00029327. DOI: 10.2307/2372398.

O'Neill, Barrett (Dec. 1966). "The Fundamental Equations of a Submersion". In: *Michigan Mathematical Journal* 13.4, pp. 459–469. ISSN: 0026-2285, 1945-2365. DOI: 10.1307/mmj/1028999604.

Paulin, Frédéric (2007). *Géométrie différentielle élémentaire.*
— (2014). *Géométrie Riemanienne, cours de second année de mastère.*
Pennec, Xavier (July 2006). "Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements". In: *Journal of Mathematical Imaging and Vision* 25.1, pp. 127–154. ISSN: 0924-9907, 1573-7683. DOI: 10.1007/s10851-006-6228-4.
Pennec, Xavier and Vincent Arsigny (2013). "Exponential Barycenters of the Canonical Cartan Connection and Invariant Means on Lie Groups". In: *Matrix Information Geometry.* Ed. by Frank Nielsen and Rajendra Bhatia. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 123–166. ISBN: 978-3-642-30231-2. DOI: 10.1007/978-3-642-30232-9_7.
Pennec, Xavier, Pierre Fillard, and Nicholas Ayache (Jan. 2006). "A Riemannian Framework for Tensor Computing". In: *International Journal of Computer Vision* 66.1, pp. 41–66. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-005-3222-z.
Pennec, Xavier, Stefan Sommer, and Tom Fletcher, eds. (2020). *Riemannian Geometric Statistics in Medical Image Analysis.* Vol. 3. The Elsevier and MICCAI Society Book Series. Elsevier. ISBN: 978-0-12-814725-2. DOI: 10.1016/C2017-0-01561-6.
Postnikov, Mikhail Mikhailovich (2001). *Geometry VI: Riemannian Geometry.* Encyclopaedia of Mathematical Sciences, Geometry. Berlin Heidelberg: Springer-Verlag. ISBN: 978-3-540-41108-6.
Riemann, Bernhard (1868). "Über Die Hypothesen, Welche Der Geometrie Zu Grunde Liegen. (Mitgetheilt Durch r. Dedekind)". In: *Abhandlungen der Königlichen Gesellschaft der Wissenschaften in Göttingen* 13, pp. 133–152.
— (May 1873). "On the Hypotheses Which Lie at the Bases of Geometry". In: *Nature* 8.184, pp. 36–37. ISSN: 1476-4687. DOI: 10.1038/008036a0.
Said, Salem, Hatem Hajri, Lionel Bombrun, and Baba C. Vemuri (Feb. 2018). "Gaussian Distributions on Riemannian Symmetric Spaces: Statistical Learning With Structured Covariance Matrices". In: *IEEE Transactions on Information Theory* 64.2, pp. 752–772. ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2713829.
Schmid, Rudolf (Jan. 2004). "Infinite Dimentional Lie Groups with Applications to Mathematical Physics". In: *Journal of Geometry and Symmetry in Physics* 1, pp. 54–120. ISSN: 1312-5192, 1314-5673. DOI: 10.7546/jgsp-1-2004-54-120.
Smirnov, Oleg (July 2021). "TensorFlow RiemOpt: A Library for Optimization on Riemannian Manifolds". In: arXiv: 2105.13921.
Terras, Audrey (1988). *Harmonic Analysis on Symmetric Spaces and Applications, Vol. II.* New York: Springer-Verlag.
Thanwerdas, Yann (May 2022). "Riemannian and stratified geometries of covariance and correlation matrices". en. PhD Thesis. Université Côte d'Azur.
Thanwerdas, Yann and Xavier Pennec (July 2021a). "Geodesics and Curvature of the Quotient-Affine Metrics on Full-Rank Correlation Matrices". In: *GSI 2021 - 5th Conference on Geometric Science of Information.* Proceedings of Geometric Science of Information. Paris, France.
— (Sept. 2021b). "O(n)-Invariant Riemannian Metrics on SPD Matrices".
Townsend, James, Niklas Koep, and Sebastian Weichwald (2016). "Pymanopt: A Python Toolbox for Optimization on Manifolds Using Automatic Differentiation". In: *Journal of Machine Learning Research* 17.137, pp. 1–5. DOI: 10.5555/2946645.3007090.
Whitney, Hassler (1936). "Differentiable Manifolds". In: *Annals of Mathematics* 37.3, pp. 645–680. ISSN: 0003-486X. DOI: 10.2307/1968482.
Wood, Andrew T. A. (1994). "Simulation of the von Mises Fisher Distribution". In: *Communications in Statistics - Simulation and Computation* 23.1, pp. 157–164. DOI: 10.1080/03610919408813161.
Wynn, Kieran (2014). *PyQuaternions: A Fully Featured, Pythonic Library for Representing and Using Quaternions.*

Yair, Or, Mirela Ben-Chen, and Ronen Talmon (Apr. 2019). “Parallel Transport on the Cone Manifold of SPD Matrices for Domain Adaptation”. In: *IEEE Transactions on Signal Processing*. Vol. 67, pp. 1797–1811. DOI: 10.1109/TSP.2019.2894801.

Zimmermann, Ralf (Jan. 2017). “A Matrix-Algebraic Algorithm for the Riemannian Logarithm on the Stiefel Manifold under the Canonical Metric”. In: *SIAM Journal on Matrix Analysis and Applications* 38.2, pp. 322–342. ISSN: 0895-4798. DOI: 10.1137/16M1074485.