Adv Comput Math (2014) 40:283–313
DOI 10.1007/s10444-013-9308-1

Optimization over geodesics for exact principal geodesic analysis

S. Sommer · F. Lauze · M. Nielsen

Received: 29 June 2012 / Accepted: 2 May 2013 /
Published online: 14 June 2013

Abstract In fields ranging from computer vision to signal processing and statistics, increasing computational power allows a move from classical linear models to models that incorporate non-linear phenomena. This shift has created interest in computational aspects of differential geometry, and solving optimization problems that incorporate non-linear geometry constitutes an important computational task. In this paper, we develop methods for numerically solving optimization problems over spaces of geodesics using numerical integration of Jacobi fields and second order derivatives of geodesic families. As an important application of this optimization strategy, we compute exact Principal Geodesic Analysis (PGA), a non-linear version of the PCA dimensionality reduction procedure. By applying the exact PGA algorithm to synthetic data, we exemplify the differences between the linearized and exact algorithms caused by the non-linear geometry. In addition, we use the numerically integrated Jacobi fields to determine sectional curvatures and provide upper bounds for injectivity radii.

Keywords Geometric optimization · Principal geodesic analysis · Manifold statistics · Differential geometry · Riemannian metrics

Mathematics Subject Classifications (2010) 65K10 · 57R99

Communicated by: J. M. Peña

S. Sommer (✉) · F. Lauze · M. Nielsen
Department of Computer Science, University of Copenhagen, Copenhagen, Denmark
e-mail: sommer@diku.dk

F. Lauze
e-mail: francois@diku.dk

M. Nielsen
e-mail: madsn@diku.dk

1 Introduction

Manifolds, sets locally modeled by Euclidean spaces, have a long and intriguing history in mathematics, and topological, differential geometric, and Riemannian geometric properties of manifolds have been studied extensively. The introduction of high performance computing in applied fields has widened the use of differential geometry, and Riemannian manifolds, in particular, are now used for modeling a range of problems possessing non-linear structure. Applications include shape modeling (complex projective shape spaces [23] and medial representations of surfaces [1, 20]), imaging (tensor manifolds in diffusion tensor imaging [7, 8, 30] and image segmentation and registration [2, 31]), and several other fields (forestry [19], human motion modeling [36, 40], information geometry and signal processing [42]).

To fully utilize the power of manifolds in applied modeling, it is essential to develop fast and robust algorithms for performing computations on manifolds, and, in particular, availability of methods for solving optimization problems is paramount. In this paper, we develop methods for numerically solving optimization problems over spaces of geodesics using numerical integration of Jacobi fields and second order derivatives of geodesic families. In addition, the approach allows numerical approximation of sectional curvatures and bounds on injectivity radii [19]. The methods apply to manifolds represented both parametrically and implicitly without preconditions such as knowledge of explicit formulas for geodesics. This fact makes the approach applicable to a range of applications, and it allows exploration of the effect of curvature on non-linear statistical methods.

To exemplify this, we consider the problem of capturing the variation of a set of manifold valued data with the Principal Geodesic Analysis (PGA, [12]) non-linear generalization of Principal Component Analysis (PCA). Until now, there has been no method for numerically computing PGA for general manifolds without linearizing the problem. Because PGA can be formulated as an optimization problem over geodesics, the tools developed here apply to computing it without discarding the non-linear structure. As a result, the paper provides an algorithm for computing exact PGA for a wide range of manifolds.

1.1 Related work

A vast body of mathematical literature describes manifolds and Riemannian structures; [5, 26] provide excellent introductions to the field. From an applied point of view, the papers [4, 22, 24, 28, 35, 39] address first-order problems such as computing geodesics and solving the exponential map inverse problem, the logarithm map. Certain second-order problems including computing Jacobi fields on diffeomorphism groups [6, 44] have been considered but mainly on limited classes of manifolds. Different aspects of numerical computation on implicitly defined manifolds are covered in [32, 33, 45], and generalizing linear statistics to manifolds has been the focus of the papers [9, 12, 19, 21, 29].

Optimization problems can be posed on a manifold in the sense that the domain of the cost function is restricted to the manifold. Such problems are extensively covered in the literature (e.g. [27, 43]). In contrast, this paper concerns optimization problems over geodesics with the complexity residing in the cost functions and not the optimization domains.

The manifold generalization of linear PCA, PGA, was first introduced in [11] but it was formulated in the form most widely used in [12]. It has subsequently been used for several applications. To mention a few, the authors in [7, 12] study variations of medial atoms, [41] uses a variation of PGA for facial classification, [34] presents examples on motion capture data, and [39] applies PGA to vertebrae outlines. The algorithm presented in [12] for computing PGA with tangent space linearization is most widely used. In contrast, [34] computes PGA as defined in [11] without approximations, exact PGA, on a particular manifold, the Lie group SO(3). The paper [38] uses the methods presented here to experimentally assess the effect of tangent space linearization on high dimensional manifolds modeling real-life data.

A recent wave of interest in manifold valued statistics has lead to the development of Geodesic PCA (GPCA, [19]) and Horizontal Component Analysis (HCA, [37]). GPCA is in many respects close to PGA but GPCA optimizes for the placement of the center point and minimizes projection residuals along geodesics. HCA builds low-dimensional orthogonal decompositions in the frame bundle of the manifold that project back to approximating subspaces in the manifold.

1.2 Content and outline

The paper presents two main contributions: (1) how numerical integration of Jacobi fields and second order derivatives can be used to solve optimization problems over geodesics; and (2) how the approach allows numerical computation of exact PGA. In addition, we use the computed Jacobi fields to numerically approximate geometric properties such as sectional curvatures. After a brief discussion of the geometric background, explicit differential equations for computing Jacobi fields and second derivatives of geodesic families are presented in Section 3. The actual derivations are performed in the appendices due to their notational complexity. In Section 4, the exact PGA algorithm is developed. We end the paper with experiments that illustrate the effect of curvature on the non-linear statistical method and with estimation of sectional curvatures and injectivity radii bounds.

The importance of curvature computations is noted in [19], which lists the ability to compute sectional curvature as a high importance open problem. The paper presents a partial solution to this problem: we discuss how sectional curvatures can be determined numerically when either a parametrization or implicit representation is available.

In the experiments, we evaluate how the differences between the exact and linearized PGA depend on the curvature of the manifold. This experiment, which to the best of our knowledge has not been made before, is made possible by the generality

of the optimization approach that makes the algorithm applicable to a wide range of manifolds with varying curvature.

2 Background

This section will include brief discussions of relevant aspects of differential and Riemannian geometry. We keep the notation close to the notation used in the book [5]; see in addition Appendix A.

2.1 Manifolds and their representations

In the sequel, M will denote a Riemannian manifold of finite dimension η. We will need M to be sufficiently smooth, i.e. of class C^k for $k = 3$ or 4 depending on the application. For concrete computational applications, we will represent M either using *parametrizations* or *implicitly*. A local parametrization is a map $\mathbf{x} \in C^k(U, M)$ from an open subset $U \subset \mathbb{R}^\eta$ to M. With an implicit representation, M is represented as a level set of a differentiable map $F : \mathbb{R}^m \to \mathbb{R}^n$, e.g. $M = F^{-1}(0)$. If the Jacobian matrix DF has full rank n everywhere on M, M will be an $(m - n)$-dimensional manifold. The space $\mathbb{R}^m$ is called the embedding space. When dealing with implicitly defined manifolds, we let m and n denote the dimension of the domain and codomain of F, respectively, so that the dimension η of the manifold equals $m - n$. Examples of applications using implicit representations include shape and human poses models [18, 39], and several shape models use parametric representations [20, 25].[1]

2.2 Geodesic systems

Given a local parametrization $\mathbf{x} : U \to M$, a curve α_t on M is a geodesic if the curve x_t in U representing α_t, i.e. $\mathbf{x}^{-1} \circ \alpha_t = x_t$, satisfies the ODE

$$\ddot{x}_t^k = -\sum_{i,j}^{\eta} \Gamma_{ij}^k(x_t)\dot{x_t}^i\dot{x_t}^j, \ k = 1, \ldots, \eta \, . \tag{1}$$

Here Γ_{ij}^k denotes the Christoffel symbols of the Riemannian metric. Conversely, geodesics can be found by solving the ODE with a starting point $x_0 = q$ and initial velocity $\dot{x}_0 = v$ as initial conditions. The exponential map $\mathrm{Exp}_q v$ maps the initial point $q \in M$ and velocity $v \in T_q M$ to α_1, the point on the geodesic at time $t = 1$. When defined, the logarithm map $\mathrm{Log}_q y$ is the inverse of Exp_q, i.e. $\mathrm{Exp}_q \mathrm{Log}_q y = y$. For implicitly represented manifolds, the classical ODE describing geodesics is not directly usable because neither parametrizations nor Christoffel symbols are directly

[1] Other representations include discrete triangulations used for surfaces and quotients $\tilde{M}/G$ of a larger manifold $\tilde{M}$ by a group G. The latter is for example the case for Kendall's shape-spaces Σ_d^k [23]. Kendall's shape-spaces for planar points are actually complex projective spaces $\mathbb{C}P^{k-2}$ for which parameterizations are available, and, for points in 3-dimensional space and higher, the shape-spaces are anomalous and not manifolds. The spaces studied in [19] belong to this class.

available. Instead, the geodesic with initial point q and initial velocity v can be found as the x-part of the solution of the IVP

$$\begin{aligned} \dot{p}_t &= -\left(\sum_{k=1}^{n} \mu^k(x_t, p_t) H_{x_t}(F^k)\right) \dot{x}_t \,, \\ \dot{x}_t &= \left(I - D_{x_t} F^\dagger D_{x_t} F\right) p_t \,, \\ x_0 &= q, \;\; p_0 = v \,, \end{aligned} \tag{2}$$

see [4]. Note that x_t is a curve in the embedding space $\mathbb{R}^m$ but since M is a subset of the embedding space and the starting point q is in M, x_t will stay in M for all t. Recall that $F : \mathbb{R}^m \to \mathbb{R}^n$ is the map defining the manifold by e.g. $M = F^{-1}(0)$ and that $H(F^k)$ denotes the Hessian of the kth component of F. F is map between Euclidean spaces and the Hessian is therefore the ordinary Euclidean Hessian matrix. The map $\mu : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^n$ is defined by $(x, p) \mapsto -(D_x F^T)^\dagger p$ where the symbol $DF^\dagger$ denotes the generalized inverse or pseudo-inverse of the non-square matrix DF. Since DF has full-rank n, $DF^\dagger$ equals $DF^T(DFDF^T)^{-1}$. Numerical stability of the geodesic system is treated in [4].

2.3 Geodesic families and variations of geodesics

In the next sections, we will treat optimization problems over geodesics of which the PGA problem (6) constitute a concrete example; in addition, problems such as geodesic regression [10] and manifold total least squares belong to this class. For this purpose, we here recall the close connection between variations of geodesics, Jacobi fields, and the differential $d\text{Exp}$. Let $\alpha_{t,s}$ be a family of geodesics parametrized by s, i.e. for each $\tilde{s}$, the curve $t \mapsto \alpha_{t,\tilde{s}}$ is a geodesic. By varying the parameter s, a vector field $\frac{d}{ds}\alpha_{t,0}$ is obtained.[2] These *Jacobi fields* are uniquely determined by the initial conditions J_0 and $\frac{D}{dt}J_0$, the variation of the initial points $x_{0,s}$ and the covariant derivative of the field at $t = 0$, respectively. Define $q_s = x_{0,s}$, $v_s = \dot{x}_{0,s}$, and $w = \frac{d}{ds}v_0$. If $\frac{d}{ds}q_0 = J_0$ and $w = \frac{D}{dt}J_0$ then $\frac{d}{ds}\text{Exp}_{q_s}(tv_s)|_{s=0}$ equals J_t [5, Chap. 5]. When q_s is constant, i.e. $q_s = q$, we have the following connection between J_t and the differential $d\text{Exp}$:

$$d_{tv_0}\text{Exp}_q\, tw = J_t. \tag{3}$$

Jacobi fields can equivalently be defined as solutions to an ODE that involves the curvature endomorphism of the manifold [5, Chap. 5]. However, the curvature endomorphism is not easily computed when the manifold is represented implicitly, and, therefore, the ODE is hard to use for computational applications in this case. In the next section, we numerically compute Jacobi fields by integrating the differential of the system (2).

[2] Recall that $\frac{d}{ds}\alpha_{t,0}$ is a shorthand for $\frac{d}{ds}\alpha_{t,s}|_{s=0}$, see Appendix A.

Jacobi fields can be used to retrieve various geometric information e.g. sectional curvature. Let J_t denote a Jacobi field along the geodesic α_t with $J_0 = 0$ and derivative $w = \frac{D}{dt} J_0$. Assume the vectors $v_0 = \dot{\alpha}_0$ and w are orthonormal. These vectors define a plane $\sigma = \text{span}\{v_0, w\}$ in $T_{\alpha_0} M$, and $K_{\alpha_0}(\sigma)$ denotes the sectional curvature of the plane σ. Because $K_{\alpha_0}(\sigma)$ occurs in a Taylor expansion of the length $\| J_t \|$, the sectional curvature can be estimated by

$$K_{\alpha_0}(\sigma) \approx \frac{6}{t^3}(t - \| J(t) \|) \tag{4}$$

for small t. Furthermore, if J_t is a non-zero Jacobi field with $J_0 = 0$ along a geodesic α_t and, for some $\tilde{t} > 0$, also $J_{\tilde{t}} = 0$ then $\alpha_{\tilde{t}}$ is called a conjugate point to α_0. This can provide an upper bound on the injectivity radius of M, that, in general terms, specifies the minimum length of non-minimizing geodesics. Figure 1 illustrates the situation on the sphere $\mathbb{S}^2$. We will explore both these points in the experiments section.

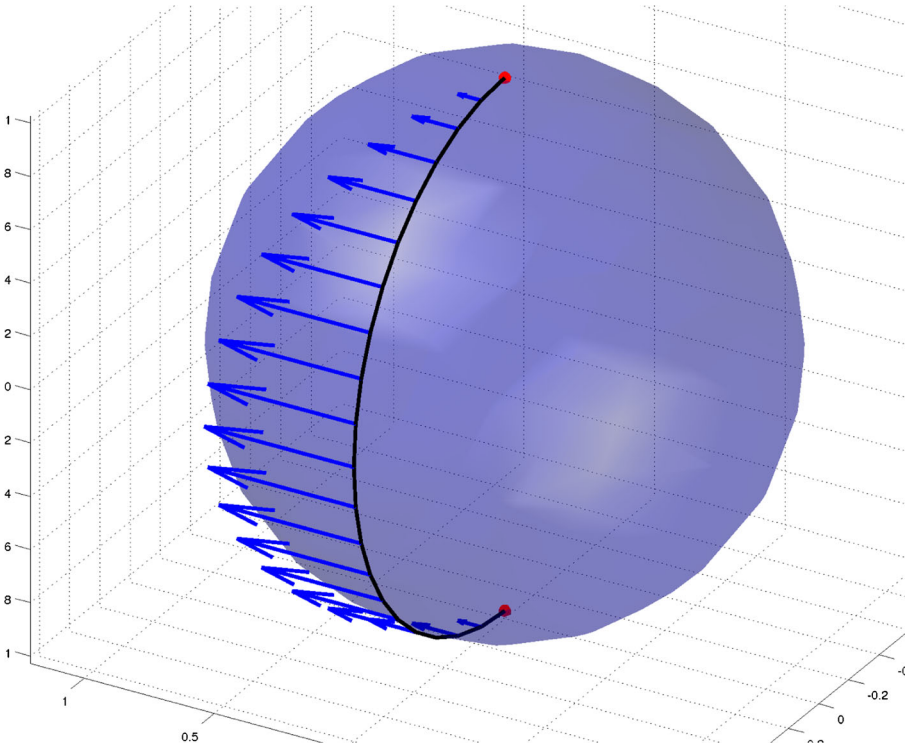


Fig. 1 The sphere $\mathbb{S}^2$ with a Jacobi field along a geodesic connecting the poles. Each pole is a conjugate point to the other since the non-zero Jacobi field vanishes. The injectivity radius is equal to the length of the geodesic, π

2.4 Principal geodesic analysis

Principal Component Analysis (PCA) is widely used to model the variability of data in Euclidean spaces. The procedure provides linear dimensionality reduction by defining a sequence of linear subspaces maximizing the variance of the projection of the data to the subspaces or, equivalently, minimizing the reconstruction errors. The kth subspace is spanned by an orthogonal basis $\{v^1, \ldots, v^k\}$ of principal components $v^1, \ldots, v^k$, and the ith principal component is defined recursively by

$$v^i = \operatorname{argmax}_{\|v\|=1} \frac{1}{N} \sum_{j=1}^{N} \left(\langle x_j, v \rangle^2 + \sum_{l=1}^{i-1} \langle x_j, v^l \rangle^2 \right) \tag{5}$$

when formulated as to maximize the variance of the projection of the dataset $\{x_1, \ldots, x_N\}$ to the subspaces span $\{v^1, \ldots, v^{i-1}\}$.

PCA is dependent on the vector space structure of the Euclidean space and hence cannot be performed on manifold valued datasets. Principal Geodesic Analysis was developed to overcome this limitation. PGA finds geodesic subspaces centered at point $\mu \in M$ with μ usually being an intrinsic mean[3] of the dataset $\{x_1, \ldots, x_N\}$, $x_j \in M$. The kth geodesic subspace S_k of $T_\mu M$ is defined as $\operatorname{Exp}_\mu(V_k)$ with $V_k = \operatorname{span}\{v^1, \ldots, v^k\}$ being the span of the principal directions $v^1, \ldots, v^k$ defined recursively by

$$\begin{aligned} v^i &= \operatorname{argmax}_{\|v\|=1, v \in V_{i-1}^\perp} \frac{1}{N} \sum_{j=1}^{N} d(\mu, \pi_{S_v}(x_j))^2, \\ S_v &= \operatorname{Exp}_\mu(\operatorname{span}\{V_{i-1}, v\}). \end{aligned} \tag{6}$$

The projection $\pi_S(x)$ of a point $x \in M$ onto a geodesic subspace $S = \operatorname{Exp}_q V$ is

$$\begin{aligned} \pi_S(x) &= \operatorname{argmin}_{y \in S} d(x, y)^2 = \operatorname{argmin}_{y \in S} \|\operatorname{Log}_y x\|^2 \\ &= \operatorname{Exp}_q(\operatorname{argmin}_{w \in V} \|\operatorname{Log}_{\operatorname{Exp}_q w} x\|^2). \end{aligned} \tag{7}$$

The term being maximized in (6) is the sample variance of the projected data, the expected value of the squared distance to μ, and PGA therefore extends PCA by finding *geodesic subspaces* in which variance is maximized.[4]

Since both optimization problems (6) and (7) are difficult to optimize, PGA has traditionally been computed using the orthogonal projection in the tangent space

[3] The notion of intrinsic mean goes back to Fréchet [13] and Karcher [21]. As in [12], an intrinsic mean is here a minimizer of $\operatorname{argmin}_{\mu \in M} \sum_{j=1}^{N} d(\mu, x_j)^2$. Uniqueness issues are treated in [21].

[4] A slightly different definition that uses one-dimensional subspaces and Lie group structure was introduced in [11].

of μ to approximate the true projection. With this approximation, equation (6) simplifies to

$$v^i \approx \text{argmax}_{\|v\|=1} \frac{1}{N} \sum_{j=1}^{N} \left(\langle \text{Log}_\mu x_j, v \rangle^2 + \sum_{l=1}^{i-1} \left\langle \text{Log}_\mu x_j, v^l \right\rangle^2 \right)$$

which is equivalent to (5), and, therefore, the procedure amounts to performing regular PCA on the vectors $\text{Log}_\mu x_j$. We will refer to PGA with the approximation as *linearized* PGA, and, following [34], PGA as defined by (6) will be referred to as *exact* PGA.[5] The ability to iteratively solve optimization problems over geodesics that we will develop in the next sections will allow us to optimize (6) and hence numerically compute exact PGA.

In general, PGA might not be well-defined as the intrinsic mean might not be unique and both existence and uniqueness may fail for the projections (7) and the optimization problem (6). The convexity bounds of Karcher [21] ensures uniqueness of the mean for sufficiently local data but setting up sufficient conditions to ensure well-posedness of (7) and (6) for general manifolds is difficult because they depend on the global geometry of the manifold.

There is ongoing discussion of when principal components should be constrained to pass the intrinsic mean as in PGA or if other types of means should be used, see [19] with discussions. In Geodesic PCA [19], the principal geodesics do not necessarily pass the intrinsic mean, and similar optimization that allows the PGA base point to move away from the intrinsic mean can be carried out with the optimization approach used in this paper. PGA can also be modified by replacing maximization of sample variance by minimization of reconstruction error. This alternate definition is not equivalent to the definition above, a fact that again underlines the difference between the Euclidean and the curved situation. We will illustrate differences between the formulations in the experiments section but we mainly use the variance formulation (6).

3 Optimization over geodesics

Equation (6) and (7) defining PGA are examples of optimization problems over geodesics that in those cases are represented by their starting point μ and initial velocity v. More generally, we here consider problems

$$\min_{(q,v)\in(M,T_qM)} F(\text{Exp}_q v) \tag{8}$$

where $F : M \to \mathbb{R}$ is a function defining the cost of the geodesic $\text{Exp}_q tv$ here at time $t = 1$.[6] In order to iteratively solve optimization problems of the form (8), we

[5] In [34], the fact that π_S has a closed form solution on the sphere $\mathbb{S}^3$ when S is a one-dimensional geodesic subspace is used to iteratively compute PGA with the [11] definition.

[6] Even more generally, F can be a function of the entire curve $\text{Exp}_q tv, t \in \mathbb{R}$ instead of just for the point $\text{Exp}_q tv, t = 1$ Note that for PGA, the initial velocity is in addition constrained to subspaces of T_qM.

will need derivatives of $\mathrm{Exp}_q v$ since $dF(\mathrm{Exp}_q v) = d_y F\, d\mathrm{Exp}_q v$ with $y = \mathrm{Exp}_q v$. Thus, we wish to compute the differential of $\mathrm{Exp}_q v$ with respect to initial point q and initial velocity v. Since (6) is a function of the projection π_S given by (7), we will later see that we need the second order differential of Exp as well.

Only in specific cases where explicit expressions for geodesics are available can the above mention differentials be derived in closed form. Instead, for general manifolds, the ODEs (1) and (2) describing geodesics can be differentiated giving systems that can be numerically integrated to provide the differentials. This approach relies on the fact that sufficiently smooth initial value problems (IVPs) are differentiable with respect to their initial values, see e.g. [16, Chap. I.14].

We will here derive explicit expressions for IVPs describing the differential of the exponential map and Jacobi fields. In addition, we will differentiate the IVPs a second time. The concrete expressions will allow the IVPs to be used for iterative optimization of problems on the form (8). In particular, they will be used for the exact PGA algorithm presented in the next section. The basic strategy is simple: we differentiate the geodesic systems of Section 2.2. Though the resulting equations are notationally complex, their derivation is in principle just repeated application of the chain and product rules for differentiation. MATLAB code for numerical integration of the systems is available at http://image.diku.dk/sommer.

Since the geodesic equations (2) contain the generalized inverse of the Jacobian matrix DF, we will use the following formula for derivatives of generalized inverses. When an $n \times m$ matrix A_s depends on a parameter s and has full rank n, and if its generalized inverse $A_s^\dagger$ is differentiable, then the derivative $\frac{d}{ds}(A_s^\dagger)$ is given by

$$\frac{d}{ds}(A_s^\dagger) = -A_s^\dagger\left(\frac{d}{ds}A_s\right)A_s^\dagger + \left(I - A_s^\dagger A_s\right)\left(\frac{d}{ds}A_s^T\right)\left(A_s^\dagger\right)^T A_s^\dagger A_s\,. \tag{9}$$

This result was derived in [3, 14] and [17] for the full-rank case. We will apply (9) with $A_s = D_{x_{t,s}}F$ when $x_{t,s}$ is an s dependent family of curves in the embedding space $\mathbb{R}^m$ that are geodesics on M and when t is fixed. To see that $D_{x_{t,s}}F^\dagger$ is differentiable with respect to s when $x_{t,s}$ depends smoothly on s, take a frame of the normal space to M in a neighborhood of $x_{t,s}$, and note that $D_{x_{t,s}}F^\dagger$ is a composition of a invertible map onto the frame depending smoothly on s and the frame itself.

The explicit expressions for the differential equations are notationally heavy. Therefore, we only state the results here and postpone the actual derivation to Appendix B.

Let $M \subset \mathbb{R}^m$ be defined as a regular zero level set of a C^3 map $F : \mathbb{R}^m \to \mathbb{R}^n$. Using the embedding, we identify curves in M and vectors in TM with curves and vectors in $\mathbb{R}^m$. Let x_t be a geodesic with $x_0 = q$ and $\dot{x}_0 = v$. The Jacobi field J_t along x_t with $J_0 = u$ and $\frac{D}{dt}J_0 = w$ can then be found as the z-part of the solution of the IVP

$$\begin{aligned}\begin{pmatrix}\dot{y}_t\\ \dot{z}_t\end{pmatrix} &= F^I_{q,v}\left(t, \begin{pmatrix}y_t\\ z_t\end{pmatrix}\right),\\ \begin{pmatrix}y_0\\ z_0\end{pmatrix} &= \begin{pmatrix}w\\ u\end{pmatrix},\end{aligned} \tag{10}$$

with $F^I_{q,v}$ the map given in explicit form in Appendix B.

As previously noted, Jacobi fields can be described using an ODE incorporating the curvature endomorphism in the parameterized case. We can, however, apply a procedure similar to the implicit case and derive and IVP by differentiating the geodesic system (1). We will use the resulting IVP (24) when working with variations of geodesics in the parameterized case, see Appendix B.

The systems (10) and (24) are both linear in the initial values $(w\ u)^T$ as expected of systems describing differentials. They are non-autonomous due to the dependence on the position on the curve x_t.

Recall the equivalence (3) between Jacobi fields and $d\text{Exp}$: if (y_t, z_t) satisfy (10) (or (24)) with initial values $(w, 0)^T$ then $d_v\text{Exp}_q w$ is equal to z_1. Therefore, we can compute the differential $d_v\text{Exp}_q$ with respect to v by numerically integrating the system using a basis $\{w^1, \ldots, w^\eta\}$ for the tangent space T_qM at $q \in M$. With initial conditions $(0, u)^T$ instead, we can similarly compute the derivative with respect to the initial point q. Note that $\text{Exp}_q \text{Log}_q y = y$ implies that $d_y\text{Log}_q = (d_{\text{Log}_q y}\text{Exp}_q)^{-1}$, a fact that allows the computation of $d_y\text{Log}_q$ as well.

Assuming the manifold is sufficiently smooth, we can differentiate the systems (10) and (24) once more and thereby obtain second order information that we will need later. The main difficulty is performing the algebra of the already complicated expressions for the systems, and, for the implicit case, we will need second order derivatives of the generalized inverses $D_{x_{t,s}} F^\dagger$. For simplicity, we consider a families of geodesics with stationary initial point. The derivations are again postponed to Appendix B.

Let M be of class C^4, and let $\alpha_{t,s}$ be a family of geodesics. Assume $\mathbf{x} : U \to M$ is a local parametrization containing $\alpha_{t,s}$, and let $x_{t,s}$ be the curve in U representing $\alpha_{t,s}$, i.e. $\mathbf{x}^{-1} \circ \alpha_{t,s} = x_{t,s}$. Let $w \in T_qM$ with $\alpha_{0,s} = q$ and $v_s = \dot{\alpha}_{0,s}$. Define $u = \frac{d}{ds} v_0$, and let $V_{q,v_0,w,u} = \frac{d}{ds}\left(d_{v_s}\text{Exp}_q w\right) = \frac{d}{ds}\left(\frac{d}{dr}\left(\text{Exp}_q v_s + r w\right)\right)$. Then, in coordinates defined by $\mathbf{x}$, $V_{q,v_0,w,u}$ can be found as the r-part of the solution of the IVP

$$\begin{aligned} \begin{pmatrix} \dot{q}_t \\ \dot{r}_t \end{pmatrix} &= G^P_{q,v_0,w,u}\left(t, \begin{pmatrix} q_t \\ r_t \end{pmatrix}\right), \\ \begin{pmatrix} q_0 \\ r_0 \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{11}$$

with $G^P_{q,v_0,w,u}$ the map given in explicit form in Appendix B.

Now, let instead $M \subset \mathbb{R}^m$ be defined as a regular zero level set of a C^4 map $F : \mathbb{R}^m \to \mathbb{R}^n$. Then $V_{q,v_0,w,u}$ can be found as the r-part of the solution of the IVP

$$\begin{aligned} \begin{pmatrix} \dot{q}_t \\ \dot{r}_t \end{pmatrix} &= G^I_{q,v_0,w,u}\left(t, \begin{pmatrix} q_t \\ r_t \end{pmatrix}\right), \\ \begin{pmatrix} q_0 \\ r_0 \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{12}$$

with $G^I_{q,v_0,w,u}$ the map given in explicit form in Appendix B.

We note that solutions to (11) and (12) depend linearly on u even though the systems themselves are not linear.

3.1 Numerical considerations

The geodesic systems (1) and (2) can in both the parametrized and implicit case be expressed in Hamiltonian forms. In [4], the authors use this property along with symplectic numerical integrators to ensure the computed curves will be close to actual geodesics. This is possible since the Hamiltonian encodes the Riemannian metric. The usefulness of pursuing a similar approach of expressing the differential systems in Hamiltonian form and using symplectic integrators to preserve the Hamiltonians is limited since there is no direct interpretation of such Hamiltonians in contrast to the case for geodesic systems.

Along the same lines, we would like to use the preservation of quadratic forms for symplectic integrators [15] to preserve quadratic properties of the differential of the exponential map, e.g. the Gauss Lemma [5]. We are currently investigating numerical schemes that could possibly ensure such stability.

4 Exact principal geodesic analysis

As an example of how the IVPs describing differentials allow optimizing over geodesics, we will provide algorithms that allow iterative optimization of (6) and that thus allow PGA as defined in [12] to be computed without the traditional linear approximation.

Solving the optimization problem (6) requires the ability to compute the projection π_S. We start with the gradient needed for iteratively computing the projection before deriving the gradient of the cost function of (6). Computing these gradients will require the differentials over geodesic families derived in Section 3. Thereafter, we present the actual algorithms for solving the problems before discussing convergence issues.

The optimization problems (6) and (7) are posed in the tangent space of the manifold at the sample mean and the unit sphere of that tangent space, respectively. These domains have relatively simple geometry, and, therefore, the complexity of the problems is contained in the cost functions. Because of this, we will not need optimizing algorithms that are specialized for domains with complicated geometry. For simplicity, we compute gradients and present steepest descent algorithms but it is straightforward to compute Jacobians instead and use more advanced optimization algorithms such as Gauss-Newton or Levenberg-Marquardt.

The overall approach is similar to the approach used for computing exact PGA in [34]. Our solution differs in that we are able to compute π_S and its differential without restricting to the manifold SO(3) and in that we optimize the functional (6) instead of the cost function used in [11] that involves one-dimensional subspaces.

4.1 The geodesic subspace projection

We consider the projection $\pi_S(x)$ of a point $x \in M$ on a geodesic subspace S. Assume S is centered at $\mu \in M$, let V be a k-dimensional subspace of $T_\mu M$ such that $S = \mathrm{Exp}_\mu V$, and define a residual function $R_{x,\mu} : V \to \mathbb{R}$ by $w \mapsto \|\mathrm{Log}_{\mathrm{Exp}_\mu w} x\|^2$ that measures squared distances between x and points in S. Computing $\pi_S(x)$ by solving (7) is then equivalent to finding $w \in V$ minimizing $R_{x,\mu}$. To find the gradient of $R_{x,\mu}$, choose an orthonormal basis for V and extend it to a basis for $T_\mu M$. Furthermore, let $w_0 \in V$ and choose an orthonormal basis for the tangent space $T_{\mathrm{Exp}_\mu w_0} M$. Karcher showed in [21] that the gradient $\mathrm{grad}^y \|\mathrm{Log}_y x\|^2$ equals $-2\mathrm{Log}_y x$, and, using this, we get the gradient of the residual function as

$$\nabla_{w_0}^{w \in V} R_{x,\mu} = -2(D_{w_0}\mathrm{Exp}_\mu)^T_{1,\ldots,k}(\mathrm{Log}_{\mathrm{Exp}_\mu w_0} x) \tag{13}$$

with $(D_{w_0}\mathrm{Exp}_\mu)_{1,\ldots,k}$ denoting the first k columns of the matrix $D_{w_0}\mathrm{Exp}_\mu$ expressed using the chosen bases.[7] This matrix can be computed using the IVPs (10) or (24).

4.2 The differential of the subspace projection

In order to optimize (6), we will need to compute gradients of the form

$$\mathrm{grad}_{v_0}^{v \in V_{v_0}^\perp} d(\mu, \pi_{S_v}(x))^2 \tag{14}$$

with $V_v = \mathrm{span}\{v^1, \ldots, v^k, v\}$, $S_v = \mathrm{Exp}_\mu(V_v)$ and $\mu \in M$.[8] This will involve the differential of $\pi_{S_v}(x)$ with respect to v. Since $\pi_{S_v}(x)$ is defined as a minimizer of (7), its differential cannot be obtained just by applying the chain and product rules. Instead, we use the implicit function theorem to define a map Ψ that equals $\pi_{S_v}(x)$ around a neighborhood of v in $T_\mu M$. We then derive the differential of Ψ.

For the result below, we extend the domain of residual function $R_{x,\mu}$ defined above from V to the entire tangent space $T_\mu M$. We will a choose basis for $T_\mu M$, and we let $H(R_{x,\mu})$ denote the Hessian matrix of $R_{x,\mu}$ with respect to the basis. Similarly, we will choose a basis for V_{v_0}, and we let $H(R_{x,\mu}|_{V_{v_0}})$ denote the Hessian matrix of $R_{x,\mu}$ restricted to V_{v_0} with respect to this basis. Using this notation, we get the following result for the derivative of the projection $\pi_{S_v}(x)$:

[7] In coordinates of the bases, the differential $d_{w_0}\mathrm{Exp}_\mu$ becomes a matrix that we write $D_{w_0}\mathrm{Exp}_\mu$. The notation $\nabla_{w_0}^{w \in V}$ denotes differentiation along the basis elements of V. See Appendix A for additional notation.

[8] Since v in (6) is restricted to the unit sphere, we will not need the gradient in the direction of v_0, and, therefore, we find the gradient in the subspace $V_{v_0}^\perp$ instead of in the larger space $\mathrm{span}\{v^1, \ldots, v^k\}^\perp$. As noted in Section 2.4, the optimization approach presented here can be extended to include optimization of the base point μ as well. Here, we use a fixed base point that for PGA is an intrinsic mean of a data set.

Proposition 1 *Let* $\{v^1, \ldots, v^k\}$ *be an orthonormal basis for a subspace* $V \subset T_\mu M$. *For each* $v \in V^\perp$, *let* V_v *be the subspace* $\operatorname{span}\{V, v\}$, *and let* $S_v = \operatorname{Exp}_\mu V_v$ *be the corresponding geodesic subspace. Fix* $v_0 \in V^\perp$ *and define* $w_0 = \operatorname{Log}_\mu \pi_{S_{v_0}}(x)$ *for an* $x \in M$. *Suppose the matrix* $H_{v_0}(R_{x,\mu}|_{V_{v_0}})$ *has full rank* $k+1$. *Extend the orthonormal basis* $\{v^1, \ldots, v^k, v_0/\|v_0\|\}$ *for* V_{v_0} *to an orthonormal basis for* $T_\mu M$. *Then*

$$\begin{aligned} D_{v_0}^{v \in V_{v_0}^\perp} \pi_{S_v}(x) \;=\; & -(D_{w_0}\operatorname{Exp}_\mu)\bar{v}_{x,\mu,v_0,S_{v_0}} \left(\nabla_{w_0}^{w \in V_{v_0}^\perp} R_{x,\mu}\right)^T \\ & + w_0^{k+1} (D_{w_0}\operatorname{Exp}_\mu) E_{x,\mu,v_0,S_{v_0}}. \end{aligned} \tag{15}$$

The coordinates of the vector $\bar{v}_{x,\mu,v_0,S_{v_0}}$ *in the basis for* V_{v_0} *are contained in the* $(k+1)$*st column of the matrix* $H_{v_0}(R_{x,\mu}|_{V_{v_0}})^{-1}$, *the scalar* w_0^{k+1} *is the* $(k+1)$*st coordinate of* w_0 *in the basis, and* $E_{x,\mu,v_0,S_{v_0}}$ *is the matrix*

$$\begin{pmatrix} -H_{w_0}\left(R_{x,\mu}|_{V_{v_0}}\right)^{-1} B_{w_0,v_0} \\ I_{\eta-(k+1)} \end{pmatrix}$$

with B_{w_0,v_0} *the last* $\eta - (k+1)$ *columns of the matrix* $(H_{w_0}\left(R_{x,\mu}\right)(V\ v_0))^T$ *and* $I_{\eta-(k+1)}$ *the identity matrix.*

Before proving the result, we discuss its use for computing the gradient (14). The assumption that the Hessian of the restricted residual $R_{x,\mu}|_{V_{v_0}}$ must have full rank is discussed below.

Because $d(\mu, \pi_{S_v}(x))^2 = \|\operatorname{Log}_\mu \pi_{S_v}(x)\|^2$, we have

$$\nabla_{v_0}^{v \in V_{v_0}^\perp} d(\mu, \pi_{S_v}(x))^2 = 2\left(\left(D_{\pi_{S_{v_0}}(x)}\operatorname{Log}_\mu\right)\left(D_{v_0}^{v \in V_{v_0}^\perp}\pi_{S_v}(x)\right)\right)^T \left(\operatorname{Log}_\mu \pi_{S_{v_0}}(x)\right), \tag{16}$$

which, combined with (15), gives (14). In order to compute the right hand side of (15), it is necessary to compute parts of the Hessian of the non-restricted residual $R_{x,\mu}$. For doing this, we will use the alternative formulation $R_{x,\mu}(w) = \|\operatorname{Log}_x \operatorname{Exp}_\mu w\|^2$ for the residual function. With $w_0, v \in T_\mu M$ let $y = \operatorname{Exp}_\mu w_0$. Working in the chosen orthonormal basis, we have

$$\nabla_{w_0} R_{x,\mu} = 2\left(\left(D_y \operatorname{Log}_x\right) D_{w_0}\operatorname{Exp}_\mu\right)^T \operatorname{Log}_x y\,.$$

and hence

$$\begin{aligned} \tfrac{d}{ds}\left(\nabla_{w_0+vs} R_{x,\mu}\right)|_{s=0} = & \; 2\left(\tfrac{d}{ds}\left(D_{\operatorname{Exp}_\mu(w_0+sv)}\operatorname{Log}_x\right)|_{s=0}\left(D_{w_0}\operatorname{Exp}_\mu\right)\right)^T \operatorname{Log}_x y \\ & + 2\left(\left(D_y \operatorname{Log}_x\right)\tfrac{d}{ds}\left(D_{w_0+vs}\operatorname{Exp}_\mu\right)|_{s=0}\right)^T \operatorname{Log}_x y \\ & + 2\left(\left(D_y \operatorname{Log}_x\right)\left(D_{w_0}\operatorname{Exp}_\mu\right)\right)^T \tfrac{d}{ds}\left(\operatorname{Log}_x \operatorname{Exp}_\mu w_0 + sv\right)|_{s=0}\,. \end{aligned} \tag{17}$$

Note that

$$\frac{d}{ds}\left(\mathrm{Log}_x \mathrm{Exp}_\mu(w_0 + sv)\right)|_{s=0} = \left(D_y \mathrm{Log}_x\right)\left(D_{w_0}\mathrm{Exp}_\mu\right) v.$$

Using that $\frac{d}{ds}\left(A_s^{-1}\right) = A_s^{-1}\left(\frac{d}{ds}A_s\right)A_s^{-1}$ for a time dependent, invertible matrix A_s[9] and the fact that $\mathrm{Exp}_x \mathrm{Log}_x z = z$ for all z, we get

$$\begin{aligned}\frac{d}{ds}\left(D_{\mathrm{Exp}_\mu(w_0+sv)}\mathrm{Log}_x\right)|_{s=0} &= \frac{d}{ds}\left(D_{\mathrm{Log}_x(\mathrm{Exp}_\mu w_0+sv)}\mathrm{Exp}_x\right)^{-1}|_{s=0}\\ &= -\left(D_y\mathrm{Log}_x\right)\frac{d}{ds}\left(D_{\mathrm{Log}_x(\mathrm{Exp}_\mu w_0+sv)}\mathrm{Exp}_x\right)|_{s=0}\left(D_y\mathrm{Log}_x\right).\end{aligned}$$

The middle term of this product and the term $\frac{d}{ds}\left(D_{w_0+sv}\mathrm{Exp}_\mu\right)|_{s=0}$ in (17) can be computed using the IVPs (11),(12) discussed in Section 3.

Proof (Proposition 1) Extend the basis $\{v^1, \ldots, v^k, v_0/\|v_0\|\}$ for V_{v_0} to an orthonormal basis for $T_\mu M$. The argument is not dependent on this choice of basis, but it will make the reasoning and notation easier. Let $S \subset T_\mu M \times V^\perp$ be an open neighborhood of (w_0, v_0) and define the map $F_V : S \to \mathbb{R}^\eta$ by

$$F_V(w,v) = \begin{pmatrix} \nabla_w R_{x,\mu}\cdot v^1 \\ \vdots \\ \nabla_w R_{x,\mu}\cdot v^k \\ \nabla_w R_{x,\mu}\cdot v \\ w\cdot u^1(v) \\ \vdots \\ w\cdot u^{\eta-k-1}(v)\end{pmatrix} = \begin{pmatrix} \begin{pmatrix} V & v\end{pmatrix}^T \nabla_w R_{x,\mu} \\ U_v^T w\end{pmatrix}$$

with the vectors $u^1(v), \ldots, u^{\eta-(k+1)}(v)$ constituting an orthonormal basis for $V_v^\perp$ for each v and with $(V\ v)$ and U_v denoting the matrices having v^i, v and $u^i(v)$ in the columns, respectively. Since $\left\langle \nabla_{w_0} R_{x,\mu}, v\right\rangle = d_{w_0}R_{x,\mu}(v) = 0$ for all $v \in V_{v_0}$ because w_0 is a minimizer for $R_{x,\mu}$ among vectors in in V_{v_0}, we see that $F_V(w_0, v_0)$ vanishes. Therefore, if $D^w_{(w_0,v_0)}F_V$ is non-singular, the implicit function theorem asserts the existence of a map Ψ from a neighborhood of v_0 to $T_\mu M$ with the property that $F_V(\Psi(v), v) = 0$ for all v in the neighborhood. We then compute

$$0 = D_{v_0}F_V(\Psi(v), v) = \left(D^w_{(w_0,v_0)}F_V\right)\left(D_{v_0}\Psi(v)\right) + \left(D^v_{(w_0,v_0)}F_V\right)$$

and hence

$$D_{v_0}^{v\in V_{v_0}^\perp}\Psi(v) = -\left(D^w_{(w_0,v_0)}F_V\right)^{-1}\left(D^{v\in V_{v_0}^\perp}_{(w_0,v_0)}F_V\right). \tag{18}$$

For the differentials on the right hand side of (18), we have

$$D^{v\in V_{v_0}^\perp}_{(w_0,v_0)}F_V = \left(0\ \cdots\ 0\ \ \nabla_{w_0}^{w\in V_{v_0}^\perp} R_{x,\mu}\ \ D^{v\in V_{v_0}^\perp}_{(w_0,v_0)}\left(w_0^T U_v\right)\right)^T$$

[9] See [3, Eq. (2)].

and

$$D^{w}_{(w_0,v_0)}F_V = \begin{pmatrix} \begin{pmatrix} V\ v_0 \end{pmatrix}^T d^{w}_{w_0}\left(\nabla_w R_{x,\mu}\right) \\ U^T_{v_0} \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} H_{w_0}\left(R_{x,\mu}\right)\begin{pmatrix} V\ v_0 \end{pmatrix} \end{pmatrix}^T \\ U^T_{v_0} \end{pmatrix}. \tag{19}$$

With the choice of basis, the above matrix is block triangular,

$$D^{w}_{(w_0,v_0)}F_V = \begin{pmatrix} A_{w_0,v_0} & B_{w_0,v_0} \\ 0 & C_{w_0,v_0} \end{pmatrix}, \tag{20}$$

with A_{w_0,v_0} equal to $H_{w_0}\left(R_{x,\mu}|_{V_{v_0}}\right)$. The requirement that $D^{w}_{(w_0,v_0)}F_V$ is non-singular is fulfilled, because $H_{w_0}\left(R_{x,\mu}|_{V_{v_0}}\right)$ has rank $k+1$ by assumption and U_{v_0} has rank $\eta-(k+1)$.

Since the first k rows of $D^{v\in V^{\perp}_{v_0}}_{(w_0,v_0)}F_V$ are zero, we need only the last $\eta-k$ columns of $\left(D^{w}_{(w_0,v_0)}F_V\right)^{-1}$ in order to compute (18). The vector $\bar{v}_{x,\mu,v_0,S_{v_0}}$ as defined in the statement of the theorem is equal to the $(k+1)$st column. Let $E_{x,\mu,v_0,S_{v_0}}$ be the matrix consisting of the remaining $\eta-(k+1)$ columns. Using the form (20), we have

$$E_{x,\mu,v_0,S_{v_0}} = \begin{pmatrix} -H_{w_0}\left(R_{x,\mu}|_{V_{v_0}}\right)^{-1} B_{w_0,v_0}C^{-1}_{w_0,v_0} \\ C^{-1}_{w_0,v_0} \end{pmatrix}.$$

Assume $\{u^1,\ldots,u^j\}$ is chosen such that $\{u^1(v_0),\ldots,u^j(v_0)\}$ equals the previously chosen basis for $V^{\perp}_{v_0}$. With this assumption, C_{w_0,v_0} is the identity matrix $I_{\eta-(k+1)}$. In addition, let w_0^{k+1} denote the $(k+1)$st component of w_0, that is, the projection of w_0 onto $v_0/\|v_0\|$. Since $w_0 \in V_{v_0}$ and by choice of U_v, Lemma 1 (see Appendix C) gives

$$D^{v\in V^{\perp}_{v_0}}_{(w_0,v_0)}\left(U_v^T w\right) = w_0^{k+1} D^{v\in V^{\perp}_{v_0}}_{(w_0,v_0)}\left(U_v^T \tfrac{v_0}{\|v_0\|}\right) = -w_0^{k+1} I_{\eta-(k+1)}.$$

Therefore,

$$D^{v\in V^{\perp}_{v_0}}_{(w_0,v_0)}F_V = \begin{pmatrix} 0 & \cdots & 0 & \nabla^{w\in V^{\perp}_{v_0}}_{w_0} R_{x,\mu} & -w_0^{k+1} I_{\eta-(k+1)} \end{pmatrix}^T. \tag{21}$$

Note, in particular, that $D^{v\in V^{\perp}_{v_0}}_{(w_0,v_0)}F_V$ is independent on the actual choice of bases U_v. Combining (18), (21), and the fact that $\bar{v}_{x,\mu,v_0,S_{v_0}}$ and $E_{x,\mu,v_0,S_{v_0}}$ constitute the needed columns of $\left(D^{w}_{(w_0,v_0)}F_V\right)^{-1}$, we get

$$D^{v\in V^{\perp}_{v_0}}_{v_0}\Psi(v) = -\bar{v}_{x,\mu,v_0,S_{v_0}}\left(\nabla^{w\in V^{\perp}_{v_0}}_{w_0} R_{x,\mu}\right)^T + w_0^{k+1} E_{x,\mu,v_0,S_{v_0}}.$$

Because $\mathrm{Exp}_{\mu}\Psi(v) = \pi_{S_v}(x)$, this provides (15). □

4.3 Exact PGA algorithm

The gradients of the cost functions enable us to iteratively solve the optimization problems (6) and (7). We let μ be an intrinsic mean of a dataset $\{x_1,\ldots,x_N\}$,

$x_i \in M$. The algorithms listed below are essentially steepest descent methods but, as previously noted, Jacobian-based optimization algorithms can be employed as well.

Algorithm 1 for computing $\pi_S(x)$ updates $w \in V$ instead of the actual point $y \in S$ that we are interested in. The vector w is related to y by $y = \mathrm{Exp}_\mu w$.

Algorithm 1 Calculate $\pi_S(x)$

Require: $x \in M$, $S = \mathrm{Exp}_\mu V$ geodesic subspace.

$w \Leftarrow$ orthogonal projection of $\mathrm{Log}_\mu x$ onto V {initial guess}
repeat

$y \Leftarrow \mathrm{Exp}_\mu w$ {vector to point}
$g \Leftarrow -2(D_{w_0}\mathrm{Exp}_\mu)^T_{1,\ldots,k}\mathrm{Log}_y x$ {gradient}
$\tilde{w} \Leftarrow w$ {previous w}
$w \Leftarrow w - g$ {update w}

until $\|\tilde{w} - w\|$ is sufficiently small.

The algorithm for solving (6) is listed in Algorithm 2. Since v in (6) is required to be on the unit sphere, the optimization will take place on a manifold, and a natural approach to compute iteration updates will use the exponential map of the sphere. Yet, because of the symmetric geometry of the sphere, we approximate this using the simpler method of adding the gradient to the previous guess and normalizing. When computing the $(k+1)st$ principal direction, we choose the initial guess as the first regular PCA vector of the data projected to $V_k^\perp$ in $T_\mu M$. See Fig. 2 for an illustration of an iteration of the algorithm.

Algorithm 2 Calculate the $(k+1)st$ principal direction of (6).

Require: $\mu, x_1, \ldots, x_N \in M$, $\{v^1, \ldots, v^k\}$ orthogonal basis for $V_k \subset T_\mu M$.

$v \Leftarrow$ first PCA vector of $\{x_j\}$ projected first to $T_\mu M$
using Log_μ and then to $V_k^\perp$ {initial guess}
repeat

$g_j \Leftarrow \nabla_v^{v \in V_v^\perp} d(\mu, \pi_{S_v}(x_j))^2$ {for each j using (16)}
$g \Leftarrow \frac{1}{N}\sum_{j=1}^N g_j$ {gradient}
$\tilde{v} \Leftarrow v$ {previous v}
$v \Leftarrow v + g$ {update v}
$v \Leftarrow v/\|v\|$ {normalize}

until $\|\tilde{v} - v\|$ is sufficiently small.

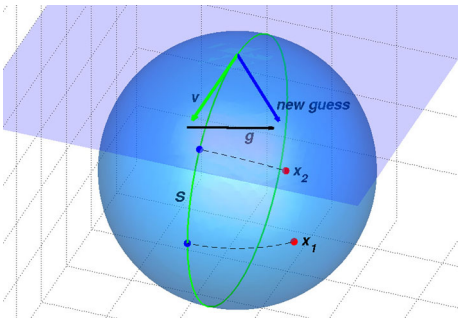


Fig. 2 An iteration of Algorithm 2. The figure shows data points x_1 and x_2 (*red points*) with projections (*blue points*) to the geodesic subspace S (*green line*). The vector v defining S is updated to the new guess by adding the gradient g and normalizing

4.4 Assumptions and convergence

As discussed in Section 2.4, because a uniqueness and existence of both the intrinsic mean and optima for (6) may fail, the PGA problem may not be well defined in itself. The uniqueness of the mean can be obtained by assuming the data is sufficiently concentrated depending on the curvature, see [21].

The curvature of the manifold may make the optimization problems non-convex, and convergence to a global optimum is therefore only ensured under the assumption that the problems (6) and (7) are convex or that no local minima exist. Giving criteria for convexity or non-existence of local optima for general manifolds and data sets is difficult because of the dependence on the global geometry of the manifolds.

The rank assumption on the Hessian used in Proposition 1 is equivalent to the residual $R_{x,\mu}$ having only non-degenerate critical points when restricted to V_{v_0}. It is shown in [21] that $R_{x,\mu}$ is convex at points sufficiently close to x and the assumption is therefore satisfied in such cases. In particular, this is satisfied if Algorithm 2 is initialized with subspaces that provide a good approximation to the data.

5 Experiments

We will use the optimization strategy and the developed algorithm for exact PGA to illustrate the differences between exact and linearized PGA. Furthermore, we will estimate sectional curvatures and compute injectivity radius bounds. Even though the algorithms are not limited to low dimensional data, we aim at visualizing the results and we will therefore provide examples with synthetic data on low dimensional manifolds. The setup allows exploring the connection between the geometry and curvature of the manifolds and the exact PGA result, and we will show how the variance and residual formulation can provide fundamentally different results. For a comparison between the methods on high dimensional manifolds modeling real-life data, we refer the reader to [38] where datasets of human vertebrae X-rays and motion capture data are treated.

The PGA algorithm is implemented in Matlab using Runge-Kutta ODE solvers. For the logarithm map, we use the shooting algorithm developed in [39]. All tolerances used for the integration and logarithm calculations are set at or lower than an

order of magnitude of the precision used for the displayed results. Intrinsic means are computed by iteratively minimizing variance using the gradient $\text{grad}^y \|\text{Log}_y x\|^2 = -2\text{Log}_y x$ (see [21]). The code used for the experiments is available at http://image.diku.dk/sommer.

We first consider surfaces embedded in $\mathbb{R}^3$ and defined by the equation

$$S_c = \left\{(x_1, x_2, x_3) | c x_1^2 + x_2^2 + x_3^2 = 1\right\}$$

for different values of the scalar c. For $c > 0$, S_c is an ellipsoid and it is equal to $\mathbb{S}^2$ in the case $c = 1$. The surface S_0 is a cylinder and, for $c < 0$, S_c is hyperboloid. Consider the point $p = (0, 0, 1)$ and note that $p \in S_c$ for all c. The curvature of S_c at p is equal to c. Note in particular that for the cylinder case the curvature is zero; the cylinder locally has the geometry of the plane $\mathbb{R}^2$ even though it informally seems to curve.

We evenly distribute 20 points along two straight lines through the origin of the tangent space $T_p S_c$, project the points from $T_p S_c$ to the surface S_c, and perform linearized and exact PGA. Figure 3 illustrates the situation in $T_p S_{-1}$ and on S_{-1} embedded in $\mathbb{R}^3$, respectively. The lines are chosen in order to ensure the points are spread over areas of the surface with different geometry. This choice is made to illustrate the influence of the curvature; a more realistic example with points sampled from a Gaussian will be provided below.

Since linearized PCA amounts to Euclidean PCA in $T_p S_c$, the first principal direction found using linearized PGA divides the angle between the lines for all c. In contrast to this, the variance and the first principal direction found using exact PGA are dependent on c. Table 1 shows the angle between the principal directions found using the two methods, the variances and variance differences for different values of c.

Let us give a brief explanation of the result. The symmetry of the sphere and the dataset cause the effect of curvature to even out in the spherical case S_1. The cylinder S_0 has local geometry equal to $\mathbb{R}^2$ which causes the equality between the methods in the $c = 0$ case. The hyperboloids with $c < 0$ that can be constructed by revolving

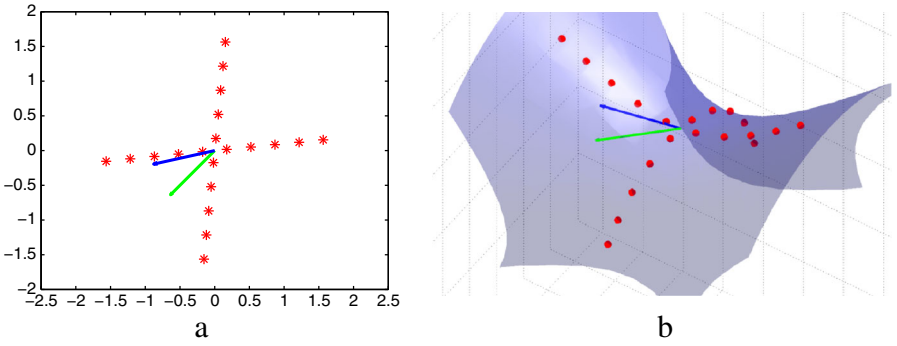


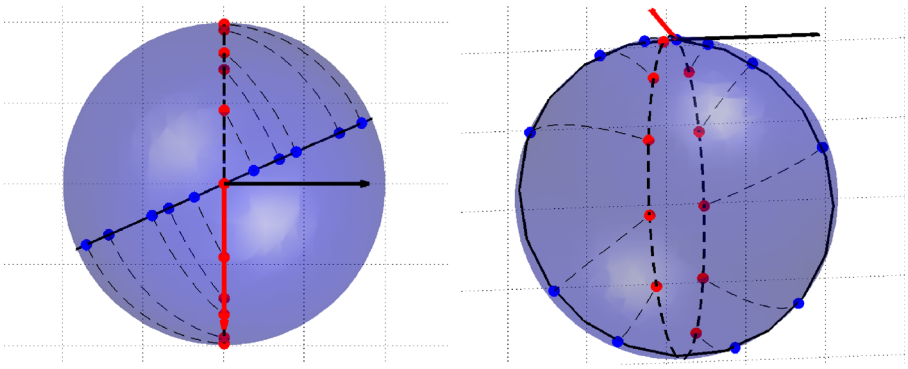
Fig. 3 The tangent space $T_p S_{-1}$ and the manifold S_{-1} with sample points. **a** $T_p S_{-1}$ with sampled points and first principal components (*blue* exact PGA, *green* linearized PGA). **b** S_{-1} with projected points and first principal components (*blue* exact PGA (6), *green* linearized PGA)

Table 1 Differences between methods for different values of c

c:	1	0.5	0	−0.5	−1	−1.5	−2	−3	−4	−5
Angle (°):	0.0	0.1	0.0	22.3	29.2	31.5	32.6	33.8	34.2	34.5
Linearized var.:	0.899	0.785	0.601	0.504	0.459	0.435	0.423	0.413	0.413	0.417
Exact var.:	0.899	0.785	0.601	0.525	0.517	0.512	0.510	0.508	0.507	0.506
Difference:	0.000	0.000	0.000	0.012	0.058	0.077	0.087	0.095	0.094	0.089
Difference (%):	0.0	0.0	0.0	4.2	12.5	17.6	20.6	23.0	22.7	21.4

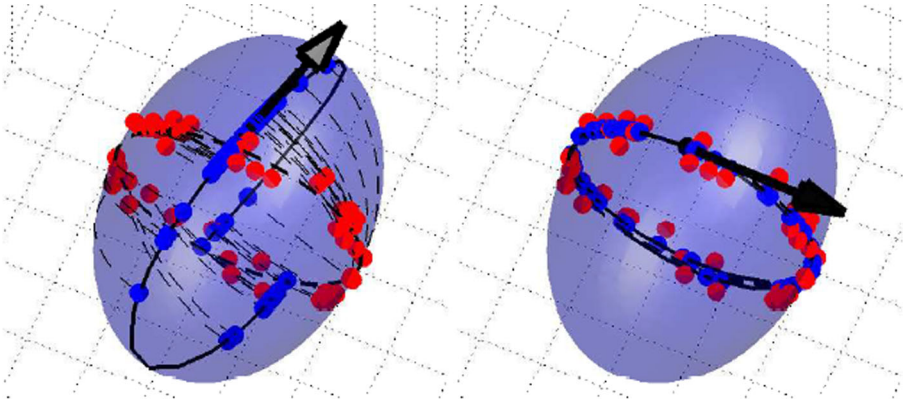
a hyperbola around its semi-minor axis are non-symmetric causing an increase in variance as the first principal direction approaches the hyperbolic axis. The effect increases with the curvature causing the first principal direction to align with the hyperbolic axis for large negative values of c. That the non-linearity is quite complex can be seen from the decreasing differences for $c = -4, -5$, a consequence of the increasing variance captured using linearized PGA. This is caused by geodesics close to the semi-minor axis being curved upwards towards the hyperbolic axis for large negative c. This results in increased captured variance that dominates the otherwise decreasing trend as c drops below -3. For all negative values of c, exact PGA is able to capture more variance in the subspace spanned by the first principal direction than linearized PGA.

Differences between the maximal variance PGA formulation (6) and the formulation that minimizes residual errors can be exemplified on simple geometries when the spread of the data is large. Similar examples for Geodesic PCA with variance formulation is reported [19]. In Fig. 4, points are sampled along a great circle through the north pole on a sphere ($c = 1$). In order to illustrate the result of maximizing projection variance, we start with the PGA center point fixed to the north pole. In this case, each iteration of the optimization procedure pushes the first principal component v^1 *away* from the direction of the great circle. In fact, the optimal direction is *orthogonal* to the direction of the great circle. This very counter-intuitive effect is caused by the projection of the points on the southern hemisphere moving closer to the south pole as the principal subspaces moves away from the great circle thus causing the measured variance to increase. In fact, the cost function (6) is non-differentiable at the optimal direction and the projections become discontinuous functions of v^1. If we instead choose the formulation that minimizes residuals, the first principal component will align with the direction of the great circle. To show that this effect persists under permutations of the data, we sample points uniformly along a geodesic on an ellipsoid ($c = 0.5$) adding Gaussian noise on the component orthogonal to the geodesic (std. dev. 0.1). This time, we optimize for the mean. The ellipsoidal geometry forces the mean to be close to the geodesic which is the reason for sampling on an ellipsoid; on a sphere, the mean is unstable under permutations of the data when the data lies close to a great circle. In Fig. 5, we show the first principal component as computed with the variance and residual formulation, respectively. As for the example on the sphere, the optimization converges to a first principal component orthogonal to the geodesic



(a) Top view of the sphere S_1. (b) Southern hemisphere of S_1.

Fig. 4 The sphere S_1 with points (*red*) sampled along a great circle (*black dotted circle*) with tangent vector (*red arrow*). The optimization for the first principal component v^1 is stopped before it reaches its optimum (25 iterations). The actual optimum (*black arrow*) is orthogonal to the great circle containing the data points. The variance is measured for the points (*blue*) projected to the current guess for the first principal subspace (*black circle*). As the guess for v^1 moves away from being tangent to the circle containing the data points, points on the southern hemisphere move southwards causing the measured variance to increase



(a) First principal component (black arrow), variance formulation (b) First principal component (black arrow), residual formulation

Fig. 5 An ellipsoid ($c = 0.5$) with points (*red*) sampled uniformly along a geodesic (*black dotted circle*). Gaussian noise (std. dev. 0.1) is added to displace the points orthogonally to the geodesic. The black arrows show the result of the optimization with (**a**) the variance formulation and (**b**) the residual formulation. With the variance formulation, the optimization is again stopped before it reaches its non-differential optimum orthogonal to the geodesic along which the points are sampled. The results show that the orthogonality of the first principal component observed in Fig. 4 also occurs with perturbed data

Table 2 Differences between the methods on M_4

Princ. comp.:	1	2	3	4
Angle (°):	10.1	10.6	12.0	12.2
Linearized var.:	1.58	3.86	4.13	4.35
Exact var.:	1.93	3.85	4.24	4.35
Difference:	0.35	−0.01	0.11	0.00
Difference (%):	21.9	−0.3	2.6	0.0

The variances of the data projected to the subspaces spanned by the first k principal directions and the percentage and angular differences are shown for $k = 1, \ldots, 4$

with the variance formulation. We again stop the optimization after a number of iterations before it reaches its non-differentiable optimum. With the residual formulation, the first principal component aligns with the geodesic along which the points are sampled. See also [19] for futher discussions on variance vs. residual formulations.

To investigate the difference between exact and linearized PGA with more than one principal direction, we consider a four dimensional manifold embedded in $\mathbb{R}^5$ and defined by

$$M_4 = \left\{(x_1, x_2, x_3, x_4, x_5) | x_1^2 - 2x_2^2 + x_3^2 - 2x_4^2 + x_5^2 = 1\right\} .$$

We make the situation more realistic than in the previous experiment by sampling 32 random points in the tangent space T_pM_4, $p = (0, 0, 0, 0, 1)$. Since T_pM_4 is an affine subspace of $\mathbb{R}^5$ orthogonal to the x_5 axis, we can identify it with $\mathbb{R}^4$ by the map $(x_1, x_2, x_3, x_4) \mapsto (x_1, x_2, x_3, x_4, 1)$. We use this identification when sampling by defining a normal distribution in $\mathbb{R}^4$, sampling the 32 points from the distribution, and mapping the results to T_pM_4. The covariance is set to $\Sigma = \text{diag}(2, 1, 2/3, 1/3)$ to get non-spherical distribution and to increase the probability of data spreading over high-curvature parts of the manifold. Table 2 lists the variances and variance differences for the four principal directions for both methods along with angular differences. The lower variance for exact PGA compared to the linearized method for the 2nd principal direction is due to the greedy definition of PGA; when maximizing variance for the 2nd principal direction, we keep the first principal direction fixed. Hence we may get lower variance than what is obtainable if we were to maximize for both principal directions together.

Table 3 Sectional curvature at p for different values of c

c:	1	0	−1	−2	−3
K_p:	1	0	−1	−2	−3
K_p est., $t = 0.01$:	1.000	0.000	−1.000	−2.000	−3.000
K_p est., $t = 0.1$:	1.000	0.000	−1.001	−2.002	−3.005

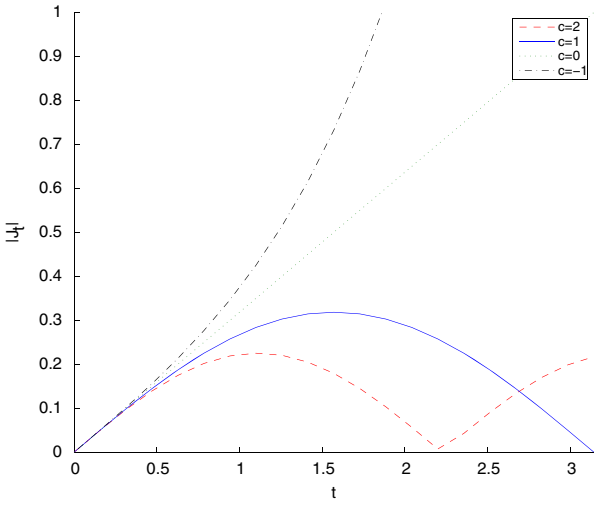


Fig. 6 $\|J_t\|$ for $c = 2, 1, 0, -1$ when $J_0 = 0$, $\frac{D}{dt}J_0 = (1, 0, 0)^T$, and $x_t = \mathrm{Exp}_p t(0, 1, 0)^T$

We clearly see angular differences between the principal directions. In addition, there is significant difference in accumulated variance in the first and third principal direction. We note that the percentage difference is calculated from what corresponds to the accumulated eigenspectrum in PCA. The percentage difference of the increase between the second and third principal direction, corresponding to the squared length of the third eigenvalue in PCA, is greater.

5.1 Curvature and conjugate points

Again considering the surfaces S_c, we can approximate the sectional curvature K_p of S_c at p using (4). The approximation is dependent on the value of the positive scalar t with increasing precision as t decreases to zero. Table 3 shows the result of the sectional curvature approximation for two values of t compared to the real sectional curvature.

Now let J_t be the Jacobi field with $J_0 = 0$ and $\frac{D}{dt}J_0 = (1, 0, 0)^T$ along the geodesic $x_t = \mathrm{Exp}_p t(0, 1, 0)^T$. Figure 6 shows $\|J_t\|$ for different values of c. We see that $\|J_\pi\| = 0$ for the spherical case S_1 showing that x_1 is a conjugate point and hence giving the upper bound π on the injectivity radius. The situation is illustrated in Fig. 1. The local geometric equivalence between the cylinder S_0 and $\mathbb{R}^2$ causes the straight line for $c = 0$. For all $c \leq 1$, the injectivity radius of S_c is π, but for $c < 1$, the point x_π is not a conjugate point.[10] By looking at $\|J_t\|$, we are only able to detect conjugate points and hence, with this experiment, we only get the bound on

[10] For $c < 1$, x_π is a *cut* point [5, Chap. 13].

the injectivity radius for $c \geq 1$. For $c > 1$ the injectivity radius decreases below 1 as seen in the case S_2 with $\|J_{\tilde{t}}\| = 0$ for $\tilde{t} \approx \pi/\sqrt{2}$.

6 Conclusion and outlook

Optimization problems over geodesics can be solved by constructing IVPs for numerical computation of Jacobi fields and second order differentials. We use this to develop an algorithm for numerically computing exact Principal Geodesic Analysis and thereby eliminating the need for the traditionally used linear approximations. In addition, the numerically computed Jacobi fields allow injectivity radii bounds and estimation of sectional curvatures partially solving an open problem stated in [19].

We use the developed algorithm to explore examples of manifold valued datasets where the principal subspaces computed by exact PGA differs from linearized PGA, and we show how the differences depend on the curvature of the manifolds and which formulation of PGA is used. In addition, we approximate sectional curvatures and bound injectivity radii and evaluate the computed results.

We are currently extending the methods to work for quotient manifolds M/G and thereby allowing the similar computations to be performed on practically all commonly occurring non-triangulated manifolds. We expect this would allow Geodesic PCA to be computed on general quotient manifolds as well. In addition, we are working on giving a theoretical treatment of the differences between the variance and residual formulations of PGA. Finally, we expect to use the automatic computation of sectional curvatures to investigate further the effect of curvature on exact PGA and other statistical methods for manifold valued data.

Acknowledgments The authors would like to thank P. Thomas Fletcher for fruitful discussions on the computation of exact PGA and Nicolas Courty for important remarks on problems regarding data locality.

Appendix A

Notation

In general, the paper follows the notation in [5]. Subscripts are used for curves on M dependent on a parameter, e.g. the curve α_t is a map $(-\epsilon, \epsilon) \to M$. The subscript notation should not be confused with differentiation with respect to the parameter t. When a local parametrization $\mathbf{x} : U \subset \mathbb{R}^\eta \to M$ is available, it is often used to represent a curve α_t so that $x_t = (x_t^1, \ldots, x_t^\eta)$ is a curve in U satisfying $\mathbf{x}^{-1} \circ \alpha_t = x_t$.

The derivative $\frac{d}{dt}\alpha_t$ of the curve α_t evaluated at $\tilde{t}$ belongs to the tangent space $T_{\alpha_{\tilde{t}}}M$. The shorthand $\frac{d}{dt}\alpha_{\tilde{t}}$ will be used for such vectors, i.e. $\frac{d}{dt}\alpha_t|_{t=\tilde{t}}$. In addition, when differentiating curves with respect to t, we often use the shorthand $\dot{\alpha}_t$. With these conventions, $\frac{d}{dt}\alpha_t|_{t=0}$, the initial velocity of the curve α_t, will be written $\dot{\alpha}_0$.

Let df denote the differential of a map $f : M \to N$ and write $d_p f$ for the differential evaluated at $p \in M$. When bases for T_pM and $T_{f(p)}M$ are specified, or when M and N are Euclidean spaces, Df is used instead of df. For maps on product manifolds, e.g. $(v, w) \mapsto g(v, w) : M \times \tilde{M} \to N$, we will need to distinguish differentiation with respect to one of the variables only. Letting one of the parameters have a fixed value w_0, the differential of the restricted function $v \mapsto g(v, w_0)$ from M to N evaluated at v_0 is denoted $d^v_{(v_0,w_0)}g$. Similarly, if V is a submanifold of M, the differential of $f|_V : V \to N$ will be denoted $d^{v\in V} f$ and its evaluation at $v_0 \in V$ will be written $d^{v\in V}_{v_0} f$.

When defined, the inverse of the exponential map Exp_q is the logarithm map denoted $\mathrm{Log}_q(\tilde{q})$. Subsets $\mathrm{Exp}_q B_r(0)$ of M with $B_r(0)$ being a ball in T_qM and with the radius $r > 0$ sufficiently small are examples of neighborhoods of q in which $\mathrm{Log}_q(\tilde{q})$ is defined. Whenever the Log-map is used, we will restrict to such neighborhoods without explicitly mentioning it.

When $h : M \to \mathbb{R}$ is a real valued function,the gradient of h with respect to the metric is denoted $\mathrm{grad}\, h$, i.e. $\mathrm{grad}\, h$ satisfies $d_p h(v) = \langle \mathrm{grad}_p h, v \rangle$ for all $v \in T_pM$. Whenever a basis of T_pM is specified, or when M is Euclidean, we switch to the usual notation ∇h. Similarly, the Hessian of h is defined by the relation $\mathrm{Hessian}(h)X = \nabla_X \mathrm{grad}\, h$ for all vector fields X using the covariant derivative ∇_X. Again, when a basis of T_pM is specified, or when M is Euclidean, the usual notation $H(h)$ will be used.

Appendix B

Expressions for the derivative ODEs

Because we will work with curves on manifolds that are either embedded in a Euclidean space or where local parametrizations are available, we can perform the derivations needed for the differential systems in Euclidean spaces: the embedding space $\mathbb{R}^m$ for the implicit case, and the parameter space $U \subset \mathbb{R}^\eta$ when a parametrization $\mathbf{x} : U \to M$ is available. The tensors we construct below will be tensors on the Euclidean spaces $\mathbb{R}^\eta$ or $\mathbb{R}^m$; they will be used as a compact notational representations, and we do not attempt to give them intrinsic geometric interpretations. The tensors will be embedding or coordinate *dependent*; this is by construction, and the tensors are thereby inherently different from intrinsic and coordinate independent tensors such as the curvature endomorphism.

The notation will as far as possible follow the tensor notation used in [5]; however, we again note that we use the embedding or parametrization to define the tensors on Euclidean domains. We will use the common identification between tensors and multilinear maps, i.e. the tensor $T : (\mathbb{R}^k)^r \to \mathbb{R}$ defines a map multilinear map $\tilde{T} : (\mathbb{R}^k)^{r-1} \to \mathbb{R}^k$ by $\left\langle \tilde{T}(y_1, \ldots, y_{r-1}), y_r \right\rangle = T(y_1, \ldots, y_r)$. We will not distinguish between a tensor and its corresponding multilinear map, and hence, in the above case, write T for both maps.

For s-dependent vector fields $v_{s,1}, \ldots, v_{s,r}$ and tensor field T_s, we will use the equality

$$\begin{aligned}&\tfrac{d}{ds} T_0(v_{0,1}, \ldots, v_{0,r}) \\ &\quad = \left(\tfrac{d}{ds} T_0\right)(v_{0,1}, \ldots, v_{0,r}) + T_0(\tfrac{d}{ds} v_{0,1}, \ldots, v_{0,r}) + \cdots + T_0\left(y_{0,1}, \ldots, \tfrac{d}{ds} v_{0,r}\right)\end{aligned} \tag{22}$$

for the derivative with respect to s. If T_{x_s} is a composition of an z-dependent tensor field T_z and an s-dependent curve x_s, the derivative $\frac{d}{ds} T_{x_s}$ equals the covariant tensor derivative $\nabla_{\frac{d}{ds} x_s} T_{x_s}$ [5, Chap. 4]. Since we will only use tensors on Euclidean spaces, such tensor derivatives will consist of component-wise derivatives.

In the following, when a parametrization $\mathbf{x}$ is available, we let T_z^P be the z-dependent 3-tensor on $\mathbb{R}^\eta$ defined by

$$T_z^P(v_1, v_2, v_3) = -\sum_{i,j,k}^{\eta} \Gamma_{ij}^k(z) v_1^i v_2^j v_3^k$$

such that the kth component of $T_{x_t}^P(\dot{x}_t, \dot{x}_t)$ equals the right hand side of (1). Note that T_z^p is symmetric in the first two components since the Christoffel symbols are symmetric in i and j. Similarly, in the implicit case, we let the z-dependent 3-tensor $T_z^{I,p}$ and 2-tensor $T_z^{I,x}$ equal the right hand side of the p and x parts of (2), respectively:

$$\begin{aligned} T_z^{I,p}(v_1, v_2) &= -\left(\sum_{k=1}^{n} \mu^k(z, v_1) H_z(F^k)\right) v_2 \,, \\ T_z^{I,x}(v) &= \left(I - D_z F^\dagger D_z F\right) v. \end{aligned}$$

The derivation below of (10) concerns the implicit case.

To derive $F_{q,v}^I$, we let $x_{t,s}$ be a family of geodesics with $x_{t,0} = x_t$, and define $q_s = x_{0,s}$ and $v_s = \dot{x}_{0,s}$. Assuming $\frac{d}{ds} q_0 = u$ and $\frac{d}{ds} v_0 = w$, the Jacobi field J_t equals $\frac{d}{ds} \mathrm{Exp}_{q_s}(t v_s)|_{s=0}$, and, therefore, we can obtain J_t by differentiating the geodesic system (2). Since M is embedded in $\mathbb{R}^m$, we consider all curves and vectors to be elements of $\mathbb{R}^m$.

We use the map μ of Section 2.2 to define the tensors

$$\begin{aligned} T_z^\mu(v) = \mu(z, v)\,, \; T_z^H(v_1, v_2) &= -\left(\textstyle\sum_{k=1}^n v_1^k H_z(F^k)\right) v_2\,, \\ T_z^D(v) = (D_z F)\, v\,, \text{ and } T_z^{D^\dagger}(v) &= (D_z F)^\dagger\, v. \end{aligned}$$

Note, in particular, that $T_z^{I,p}(v_1, v_2) = T_z^H\left(T_z^\mu(v_1), v_2\right)$. In addition, we will use the notation $\Lambda(A, B)$ for the right hand side of equation (9) so that the

derivative of a generalized inverse can be written $\frac{d}{ds}\left(A_s^\dagger\right) = \Lambda\left(A_s, \frac{d}{ds}A_s\right)$. We claim that $\frac{d}{ds}\mathrm{Exp}_{q_s}(t v_s)|_{s=0}$ equals the z-part of the solution of (10) with

$$F^I_{q,v}\left(t, \begin{pmatrix} y_t \\ z_t \end{pmatrix}\right) = \begin{pmatrix} T^{I,p}_{x_t}(p_t, \dot{z}_t) + \nabla_{z_t} T^H_{x_t}\left(T^\mu_{x_t}(p_t), \dot{x}_t\right) + T^H_{x_t}\left(T^\mu_{x_t}(y_t) - \Lambda\left(T^D_{x_t}, \nabla_{z_t} T^D_{x_t}\right)^T p_t, \dot{x}_t\right) \\ T^{I,x}_{x_t}(y_t) - \Lambda\left(T^D_{x_t}, \nabla_{z_t} T^D_{x_t}\right) T^D_{x_t}(p_t) - T^{D^\dagger}_{x_t} \nabla_{z_t} T^D_{x_t}(p_t) \end{pmatrix}. \tag{23}$$

Here $p_t = p_{t,0}$ where $p_{t,s}$ are the p-parts of the solutions to (2) with initial conditions q_s and v_s. To justify the claim, we differentiate the system (2). Using (22), we get

$$\begin{aligned} \frac{d}{dt}\frac{d}{ds}p_{t,0} &= \frac{d}{ds}\dot{p}_{t,0} = \frac{d}{ds}T^{I,p}_{x_{t,0}}(p_{t,0}, \dot{x}_{t,0}) \\ &= \nabla_{\frac{d}{ds}x_{t,0}} T^H_{x_t}\left(T^\mu_{x_t}(p_t), \dot{x}_t\right) + T^H_{x_t}\left(\nabla_{\frac{d}{ds}x_{t,0}} T^\mu_{x_t}(p_t) + T^\mu_{x_t}\left(\frac{d}{ds}p_{t,0}\right), \dot{x}_t\right) \\ &\quad + T^{I,p}_{x_t}\left(p_t, \frac{d}{ds}\dot{x}_{t,0}\right) \end{aligned}$$

and

$$\frac{d}{dt}\frac{d}{ds}x_{t,0} = \frac{d}{ds}\dot{x}_{t,0} = \frac{d}{ds}T^{I,x}_{t,0}(p_{t,0}) = \nabla_{\frac{d}{ds}x_{t,0}} T^{I,x}_{x_t}(p_t) + T^{I,x}_{x_t}\left(\frac{d}{ds}p_{t,0}\right).$$

Note that the tensor derivative $\nabla_{\frac{d}{ds}x_{t,0}} T^H_{x_t}$ consists of derivatives of $H_{x_t}(F^k)$. Both the derivatives $\nabla_{\frac{d}{ds}x_{t,0}} T^\mu_{x_t}$ and $\nabla_{\frac{d}{ds}x_{t,0}} T^{I,x}_{x_t}$ involve derivatives of generalized inverses. Therefore, we apply (9) to differentiate $T^\mu_{x_t}$ and get that

$$\nabla_{\frac{d}{ds}x_{t,0}} T^\mu_{x_t} = -\Lambda\left(T^D_{x_t}, \nabla_{\frac{d}{ds}x_{t,0}} T^D_{x_t}\right)^T.$$

The tensor derivative $\nabla_{\frac{d}{ds}x_{t,0}} T^D_{x_t}$ consists of derivatives of $D_{x_{t,s}}F$. Similarly,

$$\nabla_{\frac{d}{ds}x_{t,0}} T^{I,x}_{x_t} = -\Lambda\left(T^D_{x_t}, \nabla_{\frac{d}{ds}x_{t,0}} T^D_{x_t}\right) T^D_{x_t} - T^{D^\dagger}_{x_t} \nabla_{\frac{d}{ds}x_{t,0}} T^D_{x_t}.$$

By differentiating the initial conditions, we get (10) with $y = \frac{d}{ds}p_{t,0}$, $z = \frac{d}{ds}x_{t,0}$, and $F^I_{q,v}$ as defined in (23).

As noted, we can obtain an IVP in the parametrized case using a similar procedure. Let α_t be a geodesic in the C^3 manifold M. We assume $\mathbf{x} : U \to M$ is a local parametrization containing α_t, and we let x_t be the curve in U representing α_t, i.e. $\mathbf{x}^{-1} \circ \alpha_t = x_t$. Let $\alpha_0 = q$ and $\dot{\alpha}_0 = v$, and let u, w be vectors in $T_q M$. We associate TM with $\mathbb{R}^\eta$ using $\mathbf{x}$. The Jacobi field J_t along α_t with $J_0 = u$ and $\frac{D}{dt}J_0 = w$ can then be found as the z-part of the solution of the IVP

$$\begin{aligned} \begin{pmatrix} \dot{y}_t \\ \dot{z}_t \end{pmatrix} &= F^P_{q,v}\left(t, \begin{pmatrix} y_t \\ z_t \end{pmatrix}\right), \\ \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} &= \begin{pmatrix} w \\ u \end{pmatrix}, \end{aligned} \tag{24}$$

with $F^P_{q,v}$ the map constructed below.

To derive $F^P_{q,v}$, we let $\alpha_{t,s}$ be a family of geodesics with $\alpha_{t,0} = \alpha_t$, and define $q_s = \alpha_{0,s}$ and $v_s = \dot{\alpha}_{0,s}$. Let $x_{t,s}$ represent $\alpha_{t,s}$ using $\mathbf{x}$. Again assuming $\frac{d}{ds}q_0 = u$ and $\frac{d}{ds}v_0 = w$, we can obtain J_t by differentiating the geodesic system (1). Using (22) and symmetry of T^P_z, we have

$$\begin{aligned}\tfrac{d}{dt}\tfrac{d}{ds}x_{t,0} &= \tfrac{d}{ds}\ddot{x}_{t,0} = \tfrac{d}{ds}T^P_{x_{t,0}}(\dot{x}_{t,0}, \dot{x}_{t,0}) \\ &= \nabla_{\frac{d}{ds}x_{t,0}} T^P_{x_t}(\dot{x}_t, \dot{x}_t) + 2T^P_{x_{t,0}}\left(\tfrac{d}{dt}\tfrac{d}{ds}x_{t,0}, \dot{x}_t\right) \\ \tfrac{d}{ds}x_{0,0} &= u, \ \tfrac{d}{dt}\tfrac{d}{ds}x_{0,0} = w \end{aligned} \tag{25}$$

because $x_{t,s}$ are solutions to (1) with initial conditions q_s and v_s. Therefore, setting $y_t = \frac{d}{dt}\frac{d}{ds}x_{t,0}$ and $z_t = \frac{d}{ds}x_{t,0}$, we get (24) with

$$F^P_{q,v}\left(t, \begin{pmatrix} y_t \\ z_t \end{pmatrix}\right) = \begin{pmatrix} \nabla_{z_t} T^P_{x_t}(\dot{x}_t, \dot{x}_t) + 2T^P_{x_t}(y_t, \dot{x}_t) \\ y_t \end{pmatrix}.$$

As noted above, the derivative $\nabla_{\frac{d}{ds}x_{t,0}} T^P_{x_s}$ consists of just the component-wise derivatives of T^P_z, i.e. the derivatives of the Christoffel symbols.

For deriving the second order differentials, we will need second order derivatives of generalized inverses. Let $A_{t,s}$ be an s- and t-dependent matrix of full rank. From repeated application of the product rule and (9), we see that when the s- and t-dependent matrices $A_{t,s}$ and $A^\dagger_{t,s}$ are differentiable with respect to both variables and the mixed partial derivative $\frac{\partial^2}{\partial s \partial t}A_{t,s}$ exists then $\frac{\partial^2}{\partial s \partial t}\left(A^\dagger_{t,s}\right) = \tilde{\Lambda}\left(A_{t,s}, \frac{\partial}{\partial t}A_{t,s}, \frac{\partial}{\partial s}A_{t,s}, \frac{\partial^2}{\partial s \partial t}A_{t,s}\right)$ where

$$\begin{aligned}\tilde{\Lambda}(A, B, C, D) = &-\Lambda(A, C)BA^\dagger - A^\dagger DA^\dagger - A^\dagger B\Lambda(A, C) \\ &-\Big(\Lambda(A, C)A + A^\dagger C\Big)B^T(A^\dagger)^T A^\dagger \\ &+(I - A^\dagger A)\Big(D^T(A^\dagger)^T A^\dagger + B^T\left(\Lambda(A, C)^T A^\dagger + (A^\dagger)^T \Lambda(A, C)\right)\Big).\end{aligned} \tag{26}$$

We start the derivation with the parameterized case. We will use the tensors introduced in the beginning of this section and for the first order differentials.

We compute the q and r parts of $G^P_{q,v_0,w,u}$ separately; denote them $G^{P,q}_{q,v_0,w,u}$ and $G^{P,r}_{q,v_0,w,u}$, respectively. Let $(y^w_{t,s}, z^w_{t,s})$ be solutions to (24) with IV's $(w, 0)^T$ and along the curves $x_{t,s}$ that represents the geodesics $\alpha_{t,s}$. In addition, let y^w_t and z^w_t denote $y^w_{t,0}$ and $z^w_{t,0}$, respectively. Let also (y^u_t, z^u_t) be solutions to (24) with IV's $(u, 0)^T$ along $x_t = x_{t,0}$. Differentiating system (24), we get

$$\tfrac{d}{dt}\tfrac{d}{ds}(z^w_{t,0}) = \tfrac{d}{ds}(\dot{z}^w_{t,0}) = \tfrac{d}{ds}(y^w_{t,0})$$

and, using symmetry of the tensors,

$$\begin{aligned}\tfrac{d}{dt}\tfrac{d}{ds}(y^w_{t,0}) &= \tfrac{d}{ds}(\dot{y}^w_{t,0}) = \tfrac{d}{ds}\nabla_{z^w_{t,0}} T^P_{x_{t,0}}(\dot{x}_{t,0}, \dot{x}_{t,0}) + 2\tfrac{d}{ds}T^P_{x_{t,0}}\left(y^w_{t,0}, \dot{x}_{t,0}\right) \\ &= \nabla_{z^u_t}\nabla_{z^w_t} T^P_{x_t}(\dot{x}_t, \dot{x}_t) + \nabla_{\frac{d}{ds}z^w_{t,0}} T^P_{x_t}(\dot{x}_t, \dot{x}_t) + 2\nabla_{z^w_t} T^P_{x_t}\left(y^u_t, \dot{x}_t\right) \\ &\quad + 2\nabla_{z^u_t} T^P_{x_t}\left(y^w_t, \dot{x}_t\right) + 2T^P_{x_t}\left(\tfrac{d}{ds}y^w_{t,0}, \dot{x}_t\right) + 2T^P_{x_t}\left(y^w_t, y^u_t\right).\end{aligned} \tag{27}$$

Therefore, letting $q_t = \frac{d}{ds} y^w_{t,0}$ and $r_t = \frac{d}{ds} z^w_{t,0}$, we get $G^{P,q}_{q,v_0,w,u}(t, (r_t\ q_t)^T)$ as the right hand side of (27) and $G^{P,r}_{q,v_0,w,u}(t, (r_t\ q_t)^T)$ equal to q_t. The initial values are both 0 since $y^w_{0,s}$ and $z^w_{0,s}$ equal 0 and w, respectively, and, therefore, are not s-dependent.

For the implicit case, we will again compute the r and q parts of $G^I_{q,v_0,w,u}$ separately. Let now $\left(y^w_{t,s}, z^w_{t,s}\right)$ be solutions to (10) along the geodesics $x_{t,s}$ and with IV's $(w, 0)^T$, and let $\left(y^u_t, z^u_t\right)$ be solutions to (10) along x_t and with IV's $(u, 0)^T$. Let also $p_{t,s}$ denote the p-parts of the solutions to (2) with initial conditions q and v_s, and write $p_t = p_{t,0}$, $y^w_t = y^w_{t,0}$, and $z^w_t = z^w_{t,0}$. Recall that all curves and vectors are considered elements of the embedding space $\mathbb{R}^m$.

Differentiating system (10), we get

$$\begin{aligned}\frac{d}{dt}\frac{d}{ds} y^w_{t,0} = \frac{d}{ds}\dot{y}^w_{t,0} = \frac{d}{ds} T^{I,p}_{x_{t,0}}\left(p_{t,0}, \dot{z}^w_{t,0}\right) + \frac{d}{ds}\nabla_{z^w_{t,0}} T^H_{x_{t,0}}\left(T^\mu_{x_{t,0}}(p_{t,0}), \dot{x}_{t,0}\right)\\ + \frac{d}{ds} T^H_{x_{t,0}}\left(T^\mu_{x_{t,0}}\left(y^w_{t,0}\right) - \Lambda\left(T^D_{x_{t,0}}, \nabla_{z^w_{t,0}} T^D_{x_{t,0}}\right)^T p_{t,0}, \dot{x}_{t,0}\right) .\end{aligned}$$

Using the map $\tilde{\Lambda}$ defined in (26), we have

$$\frac{d}{ds}\Lambda\left(T^D_{x_{t,0}}, \nabla_{z^w_{t,0}} T^D_{x_{t,0}}\right)^T = \tilde{\Lambda}\left(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}, \nabla_{z^u_t}\nabla_{z^w_t} T^D_{x_t}\right)^T .$$

Combining the equations, we get

$$\begin{aligned}\frac{d}{dt}\frac{d}{ds} y^w_{t,0} = {} & \nabla_{z^u_t} T^{I,p}_{x_t}\left(p_t, \dot{z}^w_t\right) + T^{I,p}_{x_t}\left(y^u_t, \dot{z}^w_t\right) + T^{I,p}_{x_t}\left(p_t, \frac{d}{dt}\frac{d}{ds} z^w_{t,0}\right)\\
& + \nabla_{z^u_t}\nabla_{z^w_t} T^H_{x_t}\left(T^\mu_{x_t}(p_t), \dot{x}_t\right) + \nabla_{\frac{d}{ds} z^w_{t,0}} T^H_{x_t}\left(T^\mu_{x_t}(p_t), \dot{x}_t\right)\\
& + \nabla_{z^w_t} T^H_{x_t}\left(T^\mu_{x_t}(y^u_t) - \Lambda\left(T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}\right)^T p_t, \dot{x}_t\right) + \nabla_{z^w_t} T^H_{x_t}\left(T^\mu_{x_t}(p_t), \dot{z}^u_t\right)\\
& + \nabla_{z^u_t} T^H_{x_t}\left(T^\mu_{x_t}(y^w_t) - \Lambda\left(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}\right)^T p_t, \dot{x}_t\right)\\
& + T^H_{x_t}\left(T^\mu_{x_t}(\tfrac{d}{ds} y^w_{t,0}) - \Lambda\left(T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}\right)^T y^w_t, \dot{x}_t\right)\\
& - T^H_{x_t}\left(\tilde{\Lambda}\left(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}, \nabla_{z^u_t}\nabla_{z^w_t} T^D_{x_t}\right)^T p_t + \Lambda\left(T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}\right)^T y^u_t, \dot{x}_t\right)\\
& + T^H_{x_t}\left(T^\mu_{x_t}(y^w_t) - \Lambda\left(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}\right)^T p_t, \dot{z}^u_t\right) .\end{aligned}$$

Substituting $\frac{d}{ds} z^w_{t,0}$ with r_t and $\frac{d}{ds} y^w_{t,0}$ with q_t, we get $G^{I,q}_{q,v_0,w,u}$ as the right hand side of the equation. Likewise,

$$\begin{aligned}\frac{d}{dt}\frac{d}{ds} z^w_{t,0} = {} & \frac{d}{ds} T^{I,x}_{x_{t,0}}(y^w_{t,0}) - \frac{d}{ds}\Lambda\left(T^D_{x_{t,0}}, \nabla_{z^w_{t,0}} T^D_{x_{t,0}}\right) T^D_{x_{t,0}}(p_{t,0}) - \frac{d}{ds} T^{D^\dagger}_{x_{t,0}}\nabla_{z^w_{t,0}} T^D_{x_{t,0}}(p_{t,0})\\
= {} & \nabla_{z^u_t} T^{I,x}_{x_t}(y^w_t) + T^{I,x}_{x_t}\left(\tfrac{d}{ds} y^w_{t,0}\right)\\
& - \tilde{\Lambda}\left(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}, \nabla_{z^u_t}\nabla_{z^w_t} T^D_{x_t}\right) T^D_{x_t}(p_t)\\
& - \Lambda\left(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}\right)\nabla_{z^u_t} T^D_{x_t}(p_t) - \Lambda\left(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}\right) T^D_{x_t}(y^u_t)\\
& - \Lambda\left(T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}\right)\nabla_{z^w_t} T^D_{x_t}(p_t) - T^{D^\dagger}_{x_t}\nabla_{z^u_t}\nabla_{z^w_t} T^D_{x_t}(p_t) - T^{D^\dagger}_{x_t}\nabla_{z^w_t} T^D_{x_t}(y^u_t) .\end{aligned}$$

Again, after substituting $\frac{d}{ds} y^w_{t,0}$ with q_t as above, we get $G^{I,r}_{q,v_0,w,u}$ as the right hand side of the equation. As for the parametric case, both initial values are zero.

Appendix C

The projection differential

For the proof of Proposition 1, we will need the following result to show that equation (15) is independent of the chosen basis.

Lemma 1 *Let S be an open subset of $\mathbb{R}^k$ and $U : S \to M^{k\times(k-1)}$ a C^1 map with the property that for any $v \in S$, the columns of the matrix $\left(\frac{v}{\|v\|}\ U(v)\right)$ constitute an orthonormal basis for $\mathbb{R}^k$. Let u_v^j denote the jth column of $U(v)$. Then for any $v_0 \in S$ and $w \in \mathbb{R}^k$, $\left\langle \frac{d}{dt} u^j_{v_0+tw}|_{t=0}, v_0 \right\rangle = -\left\langle u^j_{v_0}, w \right\rangle$. As consequence of this, if $\tilde{U} : S \to \mathbb{R}^{k-1}$ denotes the map $v \mapsto U(v)^T \frac{v_0}{\|v_0\|}$ then*

$$D_{v_0}^{v\in \mathrm{span}\,(u^1_{v_0},\ldots,u^{k-1}_{v_0})} \tilde{U}(v) = -I_{k-1}$$

in the basis $u^1_{v_0}, \ldots, u^{k-1}_{v_0}$ for span $(u^1_{v_0}, \ldots, u^{k-1}_{v_0})$.

References

1. Blum, H., Wathen-Dunn, W.: A transformation for extracting new descriptors of shape, models for the perception of speech and visual form, 362–380 (1967)
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**, 61–79 (1995)
3. Decell, H.P.: On the derivative of the generalized inverse of a matrix. Linear and Multilinear Algebra **1**(4), 357 (1974)
4. Dedieu, J.-P., Nowicki, D.: Symplectic methods for the approximation of the exponential map and the Newton iteration on Riemannian submanifolds. J. Complex. **21**(4), 487–501 (2005)
5. do Carmo, M.P.: Riemannian geometry. In: Mathematics: Theory & Applications. Birkhauser Boston Inc., Boston (1992)
6. Ferreira, R., Xavier, J., Costeria, J., Barroso, V.: Newton algorithms for riemannian distance related problems on connected locally symmetric manifolds, Technical Report, Institute for Systems and Robotics (ISR) (2008)
7. Fletcher, P.T., Joshi, S.: Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. ECCV Workshops CVAMIA and MMBIA. **3117**, 87–98 (2004)
8. Fletcher, P.T.: Riemannian geometry for the statistical analysis of diffusion tensor data. Signal Process. **87**(2), 250–262 (2007)
9. Fletcher, P.T., Venkatasubramanian, S., Joshi, S.: Robust statistics on Riemannian manifolds via the geometric median, 2008 IEEE Conference on Computer Vision and Pattern Recognition (Anchorage, AK, USA), pp. 1–8 (2008)
10. Fletcher, P.T.: Geodesic regression on riemannian manifolds, Workshop on Mathematical Foundations of Computational Anatomy (MFCA) at MICCAI (2011)
11. Fletcher, P.T., Lu, C., Joshi, S.: Statistics of shape via principal geodesic analysis on Lie groups, CVPR 2003, vol. 1, 2003, pp. I–95–I–101 vol.1.
12. Fletcher, P.T., Lu, C., Pizer, S.M., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE Trans. Med. Imaging (2004)
13. Fréchet, M.: Les éléments aléatoires de nature quelconque dans un espace distancie. Ann. Inst. H. Poincaré **10**, 215–310 (1948)
14. Golub, G.H., Pereyra, V.: The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. SIAM J. Numer. Anal. **10**(2), 413–432 (1973)
15. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration. Springer-Verlag, Berlin (2002)

16. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I: nonstiff problems (Springer Series in Computational Mathematics), 2nd edn., Springer-Verlag, Berlin (2008)
17. Hanson, R.J., Lawson, C.L.: Extensions and applications of the householder algorithm for solving linear least squares problems. Math. Comput. **23**(108), 787–812 (1969)
18. Hauberg, S., Sommer, S., Pedersen, K.S.: Natural metrics and least-committed priors for articulated tracking. Image Vis. Comput. **30**(6–7) (2012)
19. Huckemann, S., Hotz, T., Munk, A.: Intrinsic shape analysis: geodesic PCA for Riemannian manifolds modulo isometric Lie group actions. Stat. Sin. **20**(1), 1–100 (2010)
20. Joshi, S., Pizer, S., Fletcher, P.T., Yushkevich, P., Thall, A., Marron, J.S.: Multiscale deformable model segmentation and statistical shape analysis using medial descriptions. IEEE Trans. Med. Imaging **21**(5), 538–550 (2002). PMID: 12071624
21. Karcher, H.: Riemannian center of mass and mollifier smoothing. Commun. Pur. Appl. Math. **30**(5), 509–541 (1977)
22. Keller, H.B.: Numerical methods for two-point boundary-value problems, Blaisdell, (Waltham, Mass) (1968)
23. Kendall, D.G., Manifolds, S.: Procrustean metrics and complex projective spaces. Bull. London Math. Soc. **16**(2), 81–121 (1984)
24. Klassen, E., Srivastava, A.: Geodesics between 3d closed curves using path-straightening, ECCV 2006, vol. 3951, pp. 95–106. Springer, Berlin (2006)
25. Klassen, E., Srivastava, A., Mio, W., Joshi, S.: Analysis of planar shapes using geodesic paths on shape spaces. IEEE Trans. Pattern. Anal. Mach. Intell. **26**, 372–383 (2004)
26. Lee, J.M.: Riemannian manifolds, Graduate Texts in Mathematics, vol.176, Springer-Verlag, New York (1997)
27. Luenberger, D.G.: The gradient projection method along geodesics. Manag. Sci. **18**(11), 620–631 (1972)
28. Noakes, L.: A global algorithm for geodesics. J. Aust. Math. Soc. **64**, 37–50 (1998)
29. Pennec, X.: Intrinsic statistics on Riemannian manifolds: basic tools for geometric measurements. J. Math. Imaging Vis. **25**(1), 127–154 (2006)
30. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. Int. J. Comput. Vis. **66**(1), 41–66 (2006)
31. Pennec, X., Guttmann, C., Thirion, J.-P.: Feature-based registration of medical images: Estimation and validation of the pose accuracy, MICCAI 1998, pp. 1107–1114. Springer, Berlin (1998)
32. Rabier, P.J., Rheinboldt, W.C.: On a computational method for the second fundamental tensor and its application to bifurcation problems. Numer. Math. **57**(1), 681–694 (1990)
33. Rheinboldt, W.C.: MANPAK: A set of algorithms for computations on implicitly defined manifolds. Comput. Math. Appl. **32**(12), 15–28 (1996)
34. Said, S., Courty, N., Le Bihan, N., Sangwine, S.: Exact principal geodesic analysis for data on SO(3), EUSIPCO 2007 (2007)
35. Schmidt, F., Clausen, M., Cremers, D.: Shape matching by variational computation of geodesics on a manifold, Pattern Recognition, pp. 142–151. Springer, Berlin (2006)
36. Sminchisescu, C., Jepson, A.: Generative modeling for continuous non-linearly embedded visual inference, In ICML, 759–766 (2004)
37. Sommer, S.: Horizontal dimensionality reduction and iterated frame bundle development, Geometric Science of Information (GSI) (2013)
38. Sommer, S., Lauze, F., Hauberg, S., Nielsen, M.: Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations, ECCV 2010, vol. 6316, Springer, Berlin (2010)
39. Sommer, S., Tatu, A., Chen, C., Jørgensen, D., de Bruijne, M., Loog, M., Nielsen, M., Lauze, F.: Bicycle chain shape models, MMBIA workshop at CVPR. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Miami (2009)
40. Urtasun, R., Fleet, D.J., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets, 2005 IEEE International Conference on Computer Vision (ICCV). IEEE Computer Society, pp. 403–410 (2005)
41. Wu, J., Smith, W., Hancock, E.: Weighted principal geodesic analysis for facial gender classification, progress in pattern recognition. Image Analysis and Applications, pp. 331–339. Springer, Berlin (2008)
42. Yang, L.: Means of probability measures in Riemannian manifolds and applications to radar target detection. Ph.D. thesis, Poitiers University (2011)

43. Yang, Y.: Globally convergent optimization algorithms on riemannian manifolds: uniform framework for unconstrained and constrained optimization. J. Optim. Theory Appl. **132**(2), 245–265 (2007)
44. Younes, L., Arrate, F., Miller, M.I.: Evolutions equations in computational anatomy. NeuroImage **45**(1) (2009). Supplement 1, S40–S50
45. Zhang, Q., Xu, G.: Curvature computations for n-manifolds in and solution to an open problem proposed by R. Goldman. Comput. Aided Geom. Des. **24**(2), 117–123 (2007)