

Newton Algorithms for Riemannian Distance Related Problems on Connected Locally Symmetric Manifolds

Ricardo Ferreira, *Member, IEEE*, João Xavier, *Member, IEEE*, João P. Costeira, and Victor Barroso, *Senior Member, IEEE*

Abstract—The squared distance function is one of the standard functions on which an optimization algorithm is commonly run, whether it is used directly or chained with other functions. Illustrative examples include center of mass computation, implementation of k-means algorithm and robot positioning. This function can have a simple expression (as in the Euclidean case), or it might not even have a closed form expression. Nonetheless, when used in an optimization problem formulated on non-Euclidean manifolds, the appropriate (intrinsic) version must be used and depending on the algorithm, its gradient and/or Hessian must be computed. For many commonly used manifolds a way to compute the intrinsic distance is available as well as its gradient, the Hessian however is usually a much more involved process, rendering Newton methods unusable on many standard manifolds. This article presents a way of computing the Hessian on connected locally-symmetric spaces on which standard Riemannian operations are known (exponential map, logarithm map and curvature). Although not a requirement for the result, describing the manifold as naturally reductive homogeneous spaces, a special class of manifolds, provides a way of computing these functions. The main example focused in this article is centroid computation of a finite constellation of points on connected locally symmetric manifolds since it is directly formulated as an intrinsic squared distance optimization problem. Simulation results shown here confirm the quadratic convergence rate of a Newton algorithm on commonly used manifolds such as the sphere, special orthogonal group, special Euclidean group, symmetric positive definite matrices, Grassmann manifold and projective space.

Index Terms—Riemannian distance, Hessian computation, optimization.

I. INTRODUCTION AND MOTIVATION

THE motivation behind the computation of the Hessian of the squared distance function is usually to use this important object in intrinsic Newton-like optimization methods. The advantages of these methods when compared to gradient

methods are well known, especially when high precision is required since its quadratic convergence rate is guaranteed to outperform any gradient-based algorithm when enough iterations are run. Several authors have approached the problem of implementing intrinsic Newton algorithms on smooth manifolds. For example [1], [2] and [3] discuss several implementations of Newton algorithms on manifolds and applications can be found in robotics [4], signal processing [5], image processing [6], etc.

Several approaches to the optimization of cost functions involving intrinsic squared distance exist, most of them relying on gradient methods, although there are a few exceptions where a Newton method is used. Hüper and Manton [7] have developed a Newton method for this cost function on the special orthogonal group and in [8] a Newton method which operates on an approximation of the intrinsic distance function on the Grassmann manifold. These manifolds, and others commonly used in engineering, are a subset of naturally reductive homogeneous spaces (NRHS) [9], [10]. This class of manifolds are interesting since the required operations for implementing Newton-like algorithms are easily obtained. Please note though that there is no known general relation between connected locally symmetric spaces (required for the results presented in this article) and naturally reductive homogeneous spaces (a worthy example is the Stiefel manifold which is NRHS but not locally symmetric). All the examples presented in this article: the Grassmann manifold ($\mathbb{G}(n, k)$), projective space (\mathbb{P}^n), sphere (\mathbb{S}^n), positive definite matrices ($\text{Sym}^+(n)$), the special orthogonal group ($\mathbb{SO}(n)$) and the special Euclidean group ($\mathbb{SE}(n)$) belong to the intersection of locally symmetric and naturally reductive homogeneous spaces.

The proposed application is to solve optimization problems where the cost function depends on the squared distance function using a Newton method on manifolds. In particular we provide examples for the problems of computing the centroid of a constellation of points, MAP estimation and clustering using the k-means algorithm. This article does not focus on providing state of the art implementations for these examples, but rather demonstrate how the Hessian of the intrinsic squared distance function might be used in the context of these applications. Particularly in the context of centroid computation, several authors have proposed suitable algorithms and applications, for example Moakher [11] uses centroid computation on $\mathbb{SO}(3)$ for smoothing experimental observations obtained with a significant amount of noise in the study of plate tectonics and se-

Manuscript received October 16, 2012; revised January 22, 2013 and April 19, 2013; accepted April 20, 2013. Date of publication May 06, 2013; date of current version July 15, 2013. This work was supported by the FCT project PEst-OE/EEI/LA0009/2011, partially funded by FCT Project Poeticon++ FP7-ICT-288382, FCT project Printart PTDC/EEA-CRO/098822/2008, and FCT Grant CMU-PT/SIA/0026/2009. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Nicolas Le Bihan.

The authors are with the Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon 1600-011, Portugal (e-mail: ricardo@isr.ist.utl.pt; jxavier@isr.ist.utl.pt; jpc@isr.ist.utl.pt; vab@isr.ist.utl.pt).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTSP.2013.2261799

quence-dependent continuum modeling of DNA; Manton [12] confirms the need of centroid computation algorithms on manifolds (particularly compact Lie groups); Pennec [13] uses positive definite symmetric matrices as covariance matrices for statistical characterization, also subject to smoothing; Fletcher [14] uses the computation of centroids for analyzing shapes in medical imaging. In [15] a detailed analysis of the centroid computation problem is presented as a special case of a more general problem, along with a Newton algorithm to solve it.

This article follows from two conference papers [16] and [17] where a method for computing the Hessian of the intrinsic Riemannian squared distance function on a connected locally-symmetric manifold were presented without proof. The present article is entirely self-contained with respect to the previous and consolidates them in both clarity and detail. It is important to state that most of the results required in the proof are available in the literature, and this article's role is mostly to chain them in a comprehensive way and provide a ready to use result requiring minimal knowledge of the underlying details.

II. RIEMANNIAN MANIFOLDS AND NOTATION

For a given smooth n dimensional manifold M [10], [18], [19], denote its tangent space at a point $p \in M$ by T_pM . The disjoint union of all these tangent spaces is called the tangent bundle of M and is denoted as TM . The set of real valued functions on M is $C^\infty(M)$. If M and N are smooth manifolds, given $f : M \rightarrow N$ a smooth map between them, its push-forward is defined as the application $f_* : TM \rightarrow TN$ such that for any tangent vector $X_p \in T_pM$ and any function $g \in C^\infty(N)$ the equality $f_*(X_p) \cdot g = X_p \cdot (g \circ f)$ holds. If $f \in C^\infty(M)$, exterior differentiation is denoted by df (here f is seen as a degree 0 differential form).

Additionally, the manifold M is equipped with a non-degenerate, positive and symmetric 2-tensor field g , called a Riemannian metric, providing the tangent space at each point p with an inner product $g_p : T_pM \times T_pM \rightarrow \mathbb{R}$. The notation $\langle X_p, Y_p \rangle = g_p(X_p, Y_p)$ for $X_p, Y_p \in T_pM$ will be used extensively.

The Riemannian exponential map is defined on any Riemannian manifold and sends a vector $X_q \in T_qM$ to a point on the manifold. If γ is the unique unit speed geodesic such and that $\gamma(0) = q$ and $\dot{\gamma}(0) = X_q/\|X_q\|$, then $\exp_q(X_q) = \gamma(\|X_q\|)$. In general \exp_q is only defined on a neighborhood of the origin in T_qM . However, complete spaces, defined as those where the \exp_p map has domain T_pM are very interesting in view of manifold optimization. On a sufficiently small open neighborhood this map is a diffeomorphism and the image of a ball centered at the origin contained within this neighborhood is known as a geodesic ball (or normal ball). Its inverse function known as the logarithm, when defined, returns $X_q = \log_q(p)$ such that, $\gamma(0) = q$, $\gamma(1) = p$ and $\dot{\gamma}(0) = X_q$. Although the computation of these maps may be very involved, many manifolds used in engineering have already been widely studied and these maps are usually available in the literature (see Section III-A for a simple way to compute them for the special class of naturally reductive homogeneous spaces). The length of a smooth curve $\gamma : [a, b] \rightarrow M$ is defined as $l(\gamma) = \int_a^b \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle} dt$. The

intrinsic distance between two points p, q belonging to the same connected component of M is defined as the infimum of the length of all smooth curves joining p and q .

On a Riemannian manifold there is a canonical way of identifying nearby tangent spaces called the Levi-Civita connection, here denoted by ∇ . Once a connection is established, the curvature endomorphism is defined as $R(X, Y) \cdot Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X, Y]} Z$. Here X, Y, Z are any vector fields extending $X_q, Y_q, Z_q \in T_pM$ and $[\cdot, \cdot]$ is the Lie bracket. The operator is independent of the extension chosen.

The gradient vector $\text{grad}f(p) \in T_pM$ is then defined as the unique tangent vector that satisfies $(df)_p X_p = \langle \text{grad}f(p), X_p \rangle$ for any $X_p \in T_pM$. The Hessian is defined as the symmetric 2-form such that $\text{Hess}f(q)(X_q, Y_q) = \langle \nabla_{X_q} \text{grad}f, Y_q \rangle$ for any $X_q, Y_q \in T_pM$. Note that once an orthonormal basis $\{F_{i_p}\} \subset T_pM$ for the tangent space is fixed, any tangent vector has a canonical expansion with respect to this basis and inner product given by $X_p = \sum_{i=1}^n x_i F_{i_p}$, where $x_i = \langle X_p, F_{i_p} \rangle$. These coefficients can be collected in a column matrix $\hat{X} = [x_1 \ x_2 \ \dots \ x_n]^T$, easily inputted to a computer. The hat will denote a coordinate representation for a given object on the manifold. Similarly the Hessian with respect to this basis can be described as the matrix $\hat{\mathbf{H}}$ such that $\text{Hess}f(q)(X_q, Y_q) = \hat{X}^T \hat{\mathbf{H}} \hat{Y}$ for any $X_p, Y_p \in T_pM$ with coordinate representation in this basis $\hat{X}, \hat{Y} \in \mathbb{R}^n$.

III. NEWTON'S METHOD

A. Unconstrained Optimization

Gradient descent methods (familiarity with basic optimization techniques is assumed, see for example [20] or [21] for detailed reference) are undoubtedly among the easiest to implement on smooth cost functions, as is the case of the squared distance function on \mathbb{R}^n . Unfortunately their linear convergence rate might be prohibitively expensive on applications where precision is required. Newton's method, when applicable, trades a little in implementation simplicity to gain greatly in convergence speed, guaranteeing quadratic convergence rate when close enough to the optimum.

Suppose a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is to be minimized (assume f is convex for simplicity). One way of interpreting Newton's method is to describe it as a sequence of second order Taylor expansions and minimizations. Let

$$f(x + d) \approx \hat{f}(x + d) = f(x) + \langle \text{grad}f(x), d \rangle + \frac{1}{2} d^T \mathbf{H} d$$

where \mathbf{H} is a matrix representation of the Hessian function. In \mathbb{R}^N the gradient vector field is easily computed as $\text{grad}f(x) = [(\partial/\partial x_i)f(x)]$ and the Hessian matrix has the familiar form $\mathbf{H} = [h_{ij}]$ where $h_{ij} = (\partial/\partial x_j)\text{grad}f_i(x)$. Here \hat{f} is a second order polynomial in d attaining a minimum when $d = -\mathbf{H}^{-1}\text{grad}f(x)$. The idea is that $x + d$ will be closer to the point which minimizes f .

B. Manifold Optimization

When the constraint set is a known manifold M though, the previous description still applies with only slight re-interpretations (see [2], [3] and [22] for some generalizations). A search

direction is generated $d_q \in T_q M$ as the solution of the system $\nabla_{d_q} \text{grad} f = -\text{grad} f$. If a basis for the tangent space is chosen, then the former is written as

$$\hat{\mathbf{H}} \cdot \hat{d}_q = -\hat{g} \quad (1)$$

where $\hat{\mathbf{H}}$ is a matrix representation of the Hessian of the cost function (with respect to the chosen basis for the tangent space) and \hat{g} is the representation of the gradient $\text{grad} f(q) \in T_q M$ also in the chosen basis. See Section II for a description of these intrinsic objects and Section III-A for a way of computing them in certain spaces.

As stated in the previous section, once a Newton direction has been obtained, it should be checked if it's a descent direction (its dot product with the gradient vector should be negative). If this is not verified, a fallback to the gradient descent direction should be used. Once a direction has been obtained a step in that direction must be taken. Although on a manifold it is not possible to add a vector to a point directly, a canonical way of doing it is available through the Riemannian exponential map which sends a vector $X_q \in T_q M$ to a point on the manifold as described in Section II. So the update equation $q_{k+1} = \text{exp}_{q_k}(\alpha_k d_k)$, can be used to obtain a better estimate. Here α_k is again a step size given by Armijo's rule. The complete algorithm is now described, with only slight modifications relative to the \mathbb{R}^N case:

Manifold Newton Algorithm

Input: function $f : M \rightarrow \mathbb{R}$ to be minimized.

Output: \bar{x} which minimizes f within tolerances.

1: choose $q_0 \in M$ and tolerance $\epsilon > 0$. Set $k = 0$.

2: **loop**

3: $g_k = \text{grad} f(q_k) \in T_{q_k} M$.

4: if $|g_k| < \epsilon$ set $\bar{q} = q_k$ and return.

5: compute Newton direction d_k as the solution of (1).

6: if $\langle d_k, g_k \rangle \geq 0$ set $d_k = -g_k$.

7: $\alpha_k \underset{\text{Armijo}}{\approx} \arg \min_{\alpha \geq 0} \text{exp}_{q_k}(\alpha d_k)$.

8: $q_{k+1} = \text{exp}_{q_k}(\alpha_k d_k)$. Please note that due to finite precision limitations, after a few iterations the result should be enforced to lie on the manifold.

9: $k \leftarrow k + 1$ and re-run the loop.

10: **end loop**

IV. HESSIAN OF THE RIEMANNIAN SQUARED DISTANCE FUNCTION

In [16] the following theorem was introduced without proof, and later updated in [17] still without proof. The proof is presented as an appendix to this article.

Theorem IV.1: Consider M to be a connected locally-symmetric n -dimensional Riemannian manifold with curvature endomorphism R . Let $B(p)$ be a geodesic ball centered at $p \in M$ and $r_p : B(p) \rightarrow \mathbb{R}$ the function returning the intrinsic (geodesic) distance to p . Let $\gamma : [0, r] \rightarrow B(p)$ denote the unit speed geodesic connecting p to a point $q \in B(p)$, where

$r = r_p(q)$, and let $\dot{\gamma}_q \equiv \dot{\gamma}(r)$ be its velocity vector at q . Define the function $k_p : B(p) \rightarrow \mathbb{R}$, $k_p(x) = (1/2)r_p(x)^2$ and consider any $X_q, Y_q \in T_q M$. Then

$$\text{Hess}(k_p)_q(X_q, Y_q) = \left\langle X_q^\parallel, Y_q \right\rangle + \sum_{i=1}^n \text{ctg}_{\lambda_i}(r) \langle X_q^\perp, E_{i_q} \rangle \langle Y_q, E_{i_q} \rangle. \quad (2)$$

where $\{E_{i_q}\} \subset T_q M$ is an orthonormal basis which diagonalizes the linear operator $\mathcal{R} : T_q M \rightarrow T_q M$, $\mathcal{R}(X_q) = R(X_q, \dot{\gamma}_q)\dot{\gamma}_q$ with eigenvalues λ_i , this means $\mathcal{R}(E_{i_q}) = \lambda_i E_{i_q}$. Also,

$$\text{ctg}_{\lambda}(t) = \begin{cases} \frac{\sqrt{\lambda}t}{\tan(\sqrt{\lambda}t)} & \lambda > 0 \\ 1 & \lambda = 0 \\ \frac{\sqrt{-\lambda}t}{\tanh(\sqrt{-\lambda}t)} & \lambda < 0. \end{cases}$$

Here the \parallel and \perp signs denote parallel and orthogonal components of the vector with respect to the velocity vector of γ , i.e. $X_q = X_q^\parallel + X_q^\perp$, $\langle X_q^\perp, X_q^\parallel \rangle = 0$, and $\langle X_q^\perp, \dot{\gamma}(r) \rangle = 0$.

For practical applications though, presenting the Hessian in matrix notation greatly improves its readability and comprehension. Hence in [17] a second theorem was presented also without proof which is included in the appendix as well.

Collorary IV.2: Under the same assumptions as above, consider $\{F_{i_q}\}_{i=1}^n \subset T_q M$ an orthonormal basis. If $X_q \in T_q M$ is a vector, let the notation \hat{X} denote the column vector describing the decomposition of X_q with respect to the basis $\{F_{i_q}\}$, i.e. $[\hat{X}]_i = \langle X_q, F_{i_q} \rangle$, let \mathbf{R}_k be the matrix with entries $[\mathbf{R}_k]_{ij} = \langle F_{i_q}, \mathcal{R}(F_{j_q}) \rangle$ and consider the eigenvalue decomposition $\mathbf{R}_k = E \Lambda E^T$. Here λ_i will be used to describe the i 'th diagonal element of Λ . Then the Hessian matrix (a representation for the bilinear Hessian tensor on the finite dimensional tangent space with respect to the fixed basis) is given by $\mathbf{H}_{k_p} = \mathbf{E} \Sigma \mathbf{E}^T$ where Σ is diagonal with elements σ_i given by $\sigma_i = \text{ctg}_{\lambda_i}(r)$. Hence $\text{Hess}(k_p)_q(X_q, Y_q) = \hat{X}^T \hat{\mathbf{H}}_{k_p} \hat{Y}$.

In spaces of constant curvature (such as the sphere and $\mathbb{S}\mathbb{O}(3)$) with sectional curvature λ , computing the Hessian has almost zero cost. Due to the symmetries of the curvature tensor, $\langle X_q, \mathcal{R}(Y_q) \rangle = 0$ whenever X_q or Y_q are parallel to $\dot{\gamma}_q$. Hence, matrix \mathbf{R}_k , which is the matrix representation on the given basis for the bilinear operator $\langle X_p, \mathcal{R}(Y_p) \rangle$, has a null eigenvalue with eigenvector $\dot{\gamma}_q$. Since the sectional curvature is by definition equal to $\langle X_q, \mathcal{R}(X_q) \rangle$ and is constant, equal to λ whenever X_q is not parallel to $\dot{\gamma}_q$, then

$$\left\{ \begin{array}{l} \max \\ \text{s.t.} \end{array} \left\{ \begin{array}{l} \hat{X}^T \mathbf{R}_k \hat{X} \\ \langle \hat{X}, \hat{X} \rangle = 1 \\ \langle \hat{X}, \dot{\gamma}_g \rangle = 0 \end{array} \right\} = \left\{ \begin{array}{l} \min \\ \text{s.t.} \end{array} \left\{ \begin{array}{l} \hat{X}^T \mathbf{R}_k \hat{X} \\ \langle \hat{X}, \hat{X} \rangle = 1 \\ \langle \hat{X}, \dot{\gamma}_g \rangle = 0 \end{array} \right\} \right\} = \lambda$$

hence using the Rayleigh quotient, the eigenvalues of \mathbf{R}_k are constant and equal to λ . So an eigenvalue decomposition is

$$\mathbf{R}_k = \begin{bmatrix} \dot{\gamma}_g & \dot{\gamma}_g^\perp \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \lambda \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\gamma}_g^T \\ \dot{\gamma}_g^{\perp T} \end{bmatrix}$$

where $\hat{\gamma}_q^\perp$ is any orthonormal complement of $\hat{\gamma}_q$. It follows then from the last theorem that the Hessian is given by

$$\begin{aligned} \mathbf{H}_k &= \hat{\gamma}_g \hat{\gamma}_g^T + \text{ctg}_\lambda(r) \hat{\gamma}_q^\perp \hat{\gamma}_g^{\perp T} \\ &= \hat{\gamma}_g \hat{\gamma}_g^T + \text{ctg}_\lambda(r) (\mathbf{I} - \hat{\gamma}_g \hat{\gamma}_g^T) \\ &= \text{ctg}_\lambda(r) \mathbf{I} + (1 - \text{ctg}_\lambda(r)) \hat{\gamma}_g \hat{\gamma}_g^T \end{aligned}$$

This removes the need for the numerical computation of matrix \mathbf{R}_k and its eigenvalue decomposition, significantly speeding the computation of the Hessian matrix.

A. Algorithm

The complete algorithm is presented in both situations, when the space is not known to be of constant curvature:

Hessian of Riemannian squared distance function

Input: an orthonormal base $\{F_{i_q}\} \subset T_q M$, $\hat{\gamma}_q = \log_q(p)$ and the Riemannian curvature tensor.

Output: $\hat{\mathbf{H}}$ the Hessian matrix of the Riemannian squared distance function $(1/2)r_p(q)^2$.

- 1: Build matrix $[\mathbf{R}]_{ij} = \langle F_{i_q}, \mathcal{R}(F_{j_q}) \rangle$.
- 2: Compute its eigenvalue decomposition $\mathbf{R} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T$.
- 3: Assemble diagonal matrix Σ with elements $\sigma_i = \text{ctg}_{\lambda_i}(r)$.
- 4: $\hat{\mathbf{H}} = \mathbf{E}\Sigma\mathbf{E}^T$.

or when it is known to be of constant curvature:

Hessian of Riemannian squared distance function (spaces of constant curvature)

Input: an orthonormal base $\{F_{i_q}\} \subset T_q M$, $\hat{\gamma}_q = \log_q(p)$ and the Riemannian curvature tensor.

Output: $\hat{\mathbf{H}}$ the Hessian matrix of the Riemannian squared distance function $(1/2)r_p(q)^2$.

- 1: Represent $\hat{\gamma}_q$ in the given basis.
- 2: $\hat{\mathbf{H}} = \text{ctg}_\lambda(r) \mathbf{I} + (1 - \text{ctg}_\lambda(r)) \hat{\gamma}_q \hat{\gamma}_q^T$.

V. MANIFOLD APPLICATIONS

A. Centroid Computation

Let M be a connected manifold and $\mathcal{X} = \{p_1, \dots, p_L\} \subset M$ a constellation of L points. Let $r : M \times M \rightarrow \mathbb{R}$ be the function that returns the intrinsic distance of any two points on the manifold and define a cost function $C_{\mathcal{X}} : M \rightarrow \mathbb{R}$ as

$$C_{\mathcal{X}}(q) = \frac{1}{2} \sum_{l=1}^L r(p_l, q)^2 = \sum_{l=1}^L k_{p_l}(q), \quad (3)$$

The set of solutions to the optimization problem $m_f(\mathcal{X}) = \arg \min_{q \in M} C_{\mathcal{X}}(q)$ is defined as the Fréchet mean set of the constellation and each member will be called a centroid of \mathcal{X} . Depending on the manifold M , the centroid might not be unique, for example if the sphere is considered with a constellation consisting of two antipodal points, all the equator points are centroids. The set of points at which the function (3) attains a local minimum is called the Karcher mean set and is denoted as $m_k(\mathcal{X})$. The objective is to find a centroid for the given constellation (which in the applications of interest should be unique), but the possibility of convergence to a local minimum

is not dealt with. Conditions for uniqueness of Karcher-Fréchet means usually involves the concept of manifold injectivity radius and the diameter of the constellation. Please see [12], [15], [23], [24] for the explicit treatment of these points.

Using linearity of the gradient and the Hessian operators (meaning in particular that if $f, g : M \rightarrow \mathbb{R}$ then $\text{Hess}(f+g) = \text{Hess}f + \text{Hess}g$ and $\text{grad}(f+g) = \text{grad}f + \text{grad}g$), the cost function in (3) allows for the decomposition

$$\begin{aligned} \text{grad}C_{\mathcal{X}}(q) &= \sum_{l=1}^L \text{grad}k_{p_l}(q) = - \sum_{l=1}^L \log_q(p_l) \\ \text{Hess}C_{\mathcal{X}}(q) &= \sum_{l=1}^L \text{Hess}k_{p_l}(q), \end{aligned} \quad (4)$$

where the fact that the gradient of the squared Riemannian distance function is the symmetric of the Riemannian \log map is used (as stated in [25] as a corollary to Gauss's lemma).

The algorithm for centroid computation is then

Centroid computation

Input: A constellation $\mathcal{X} = \{p_1, \dots, p_L\} \subset M$ with L points sufficiently close (see text for details) and an initial estimate q_0

Output: An element q of the Karcher mean set of the constellation

- 1: Apply Newton's algorithm as described in Section III to function $C_{\mathcal{X}}(q)$ where at each step the Hessian and gradient is computed as follows:
 - 2: **for** each point p_l in the constellation **do**
 - 3: $\text{grad}k_{p_l}(q) = -\log_q(p_l)$
 - 4: $\text{Hess}k_{p_l}(q)$ (as described in Section IV)
 - 5: **end for**
 - 6: $\text{grad}C_{\mathcal{X}}(q) = \sum_{l=1}^L \text{grad}k_{p_l}(q)$
 - 7: $\text{Hess}C_{\mathcal{X}}(q) = \sum_{l=1}^L \text{Hess}k_{p_l}(q)$

B. K-Means Algorithm

The implementation of a K-means algorithm is straightforward once a working centroid computing algorithm is available. The algorithm is as follows:

k-means algorithm

Input: An n dimensional manifold M where the centroid is computable, \mathcal{X} a cloud of points and l the number of desired classes.

Output: $\{p_1 \dots p_l\}$ centroids of each class.

- 1: Choose $\{p_1 \dots p_l\} \subset \mathcal{X}$ randomly as initial estimate for the centroids.
- 2: **for** each point q in \mathcal{X} **do**
- 3: Compute distance to each centroid: $r_i = r_q^2(p_i)$.
- 4: Label point as belonging to set \mathcal{X}_j , where $j = \arg \min_i \{r_i\}$.
- 5: **end for**
- 6: Recompute the centroids $p_i \leftarrow \text{centroid}(\mathcal{X}_i)$.
- 7: If the centroids did not change position (or a maximum number of iterations reached), return.

C. MAP Estimator

Consider a freely moving agent in \mathbb{R}^n whose position is represented as a point $T \in n$, seen as the rigid transformation that transforms points in the world referential, taking them to the local (agent) referential. Keeping the experiment simple, consider that the it observes several known landmarks in the world $\{x_1, \dots, x_k\} \in \mathbb{R}^n$. Hence, in the local referential, the agent observes the points Tx_i . If the agent is considered to be at T_0 with a certain uncertainty, it is possible to build a prior knowledge probability density function as $p(T) = k_1 \exp^{-(1/2)r^2(TT_0)/\sigma^2}$ where k_1 is a normalizing constant and σ^2 describes an isotropic level of uncertainty. Notice that all directions are treated equally which is usually not the case. Please note that by the identity $\mathbb{S}\mathbb{E}(n) = \mathbb{S}\mathbb{O}(n) \times \mathbb{R}^n$ a slightly more useful prior may be built weighting differently translations from rotations. With simplicity in mind, assume that this description is useful. Assume also that the sensors are not perfect and the observations obey the following Gaussian probability distribution $p(y_i|T) = k_2 \exp^{-(y_i - Tx_i)^T R^{-1}(y_i - Tx_i)}$ where, again k_2 is a normalizing constant and R is a matrix encoding the uncertainty of the sensor. If the observations are considered to be independent, the MAP estimator of the position is given by

$$\begin{aligned} T^* &= \arg \max_{T \in \mathbb{S}\mathbb{E}(n)} p(T|y_1, y_2, \dots, y_k) \\ &= \arg \max_{T \in \mathbb{S}\mathbb{E}(n)} \left(\prod_{i=1}^k p(y_i|T) \right) p(T) \end{aligned}$$

Using the usual trick of applying the logarithm and discarding constants, the former problem is equivalent to

$$\begin{aligned} T^* &= \arg \max_{T \in \mathbb{S}\mathbb{E}(n)} \sum_{i=1}^k -(y_i - Tx_i)^T R^{-1}(y_i - Tx_i) \\ &\quad - \frac{1}{2} r^2(T, T_0) / \sigma^2 \end{aligned}$$

This is formulated as an optimization problem on $\mathbb{S}\mathbb{E}(n)$. The gradient of each term is readily available and the Hessian of the first terms can be obtained using standard techniques (see the chapter of Riemannian Embeddings on any Riemannian geometry book, specifically the part about the second fundamental form). The result presented in this paper allows for the Hessian of the last term to be obtained as well, thus allowing for a Newton algorithm to be implemented.

VI. RESULTS

This section holds experimental results for the main application of centroid computation. Fig. 1 compares the results of applying a Newton algorithm and a standard gradient algorithm when computing the centroid of a constellation on 6 different manifolds. The 20-point constellations were generated using a radius of $\pi/3$ except for the Grassman where the radius used was $\pi/6$. The results presented in logarithmic scale clearly show the quadratic convergence rate of Newton's method and the linear convergence rate of the gradient method. All examples show a plateau at 10^{-15} due to finite precision issues. Note that the projective space manifold $\mathbb{P}^n = \mathbb{G}(n+1, 1)$ is a special case of the Grassman, hence the previous expressions are applicable.

Fig. 2 shows the results obtained for an implementation of the MAP estimator described in Section V-C. As described, an agent

is navigating in a world with 5 randomly placed landmarks. In this experiment $\sigma = 1$ and $R = I$ was used. The gradient method is clearly outperformed by the 5 iterations taken by the Newton method to attain the required precision.

As a final note concerning absolute time to convergence, none of the results of the Newton method shown in Fig. 1 are competitive with the gradient alternative. Although more iterations are required, these are simpler and thus convergence to the required precision is faster. This no longer holds for an implementation of the MAP estimator where the gradient method shows extremely slow convergence whereas Newton's method still converges typically in less than 10 iterations and outperforms the gradient implementation. The choice of method does not have a simple answer in engineering applications and in the end the actual cost function, manifold and desired precision play crucial roles. Also, as mentioned in the discussion after the statement of corollary IV.2 for certain manifolds, e.g. $\mathbb{S}\mathbb{O}(3)$, computation of the Hessian is almost trivial providing an additional advantage to Newton's method.

VII. CONCLUSIONS

This article describes a simple algorithm to obtain the Hessian of the intrinsic squared distance function on connected locally-symmetric manifolds on which it is known how to compute basic Riemannian differential operations. Results are presented for centroid computation on the commonly used manifolds $\mathbb{S}\mathbb{O}(n)$, $Sym^+(n)$, \mathbb{S}^n , $\mathbb{S}\mathbb{E}(n)$, $\mathbb{G}(n, p)$, and \mathbb{P}^n . This is by no means an exhaustive list, and the result is valid for other manifolds fitting the requisites (for example the hyperbolic plane). Besides the main application, simple examples of MAP estimation and k-means clustering are also provided, extending the range of applications besides centroid computation.

APPENDIX A

NATURALLY REDUCTIVE HOMOGENEOUS SPACES

Although this section is not critical for presenting the main result in this article, it does show that the method presented is viable by providing a recipe for obtaining the required data in a vast class of manifolds used in engineering. Note that a basic understanding of Lie group theory is assumed. Naturally Reductive Homogeneous Spaces, henceforth denoted by NRHS, (see for example [9] and [10]), are important since they can lead to closed formula solutions for the Riemannian exponential maps, logarithm maps and curvature endomorphisms, exactly what's needed to implement the Hessian algorithm presented. A space with this property is defined as a coset manifold $M = G/H$, where G is a Lie group (with Lie algebra \mathfrak{g}) and H a closed subgroup (with Lie sub-algebra $\mathfrak{h} \subset \mathfrak{g}$), furnished with a G -invariant metric such that there exists an $Ad_H(\cdot)$ invariant subspace $\mathfrak{m} \subset \mathfrak{g}$ that is complementary to $\mathfrak{h} \subset \mathfrak{g}$. Note that $\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{m}$ but \mathfrak{m} is usually not a Lie sub-algebra since it is usually not closed under the Lie bracket operation. Furthermore, the property

$$\langle [X, Y]_{\mathfrak{m}} Z \rangle = \langle X, [Y, Z]_{\mathfrak{m}} \rangle \quad \text{for } X, Y, Z \in \mathfrak{m}$$

needs to hold. Here the subscript \mathfrak{m} denotes projection on this subspace. Henceforth, for spaces with this property, \mathfrak{m} will be called a Lie subspace for G/H .

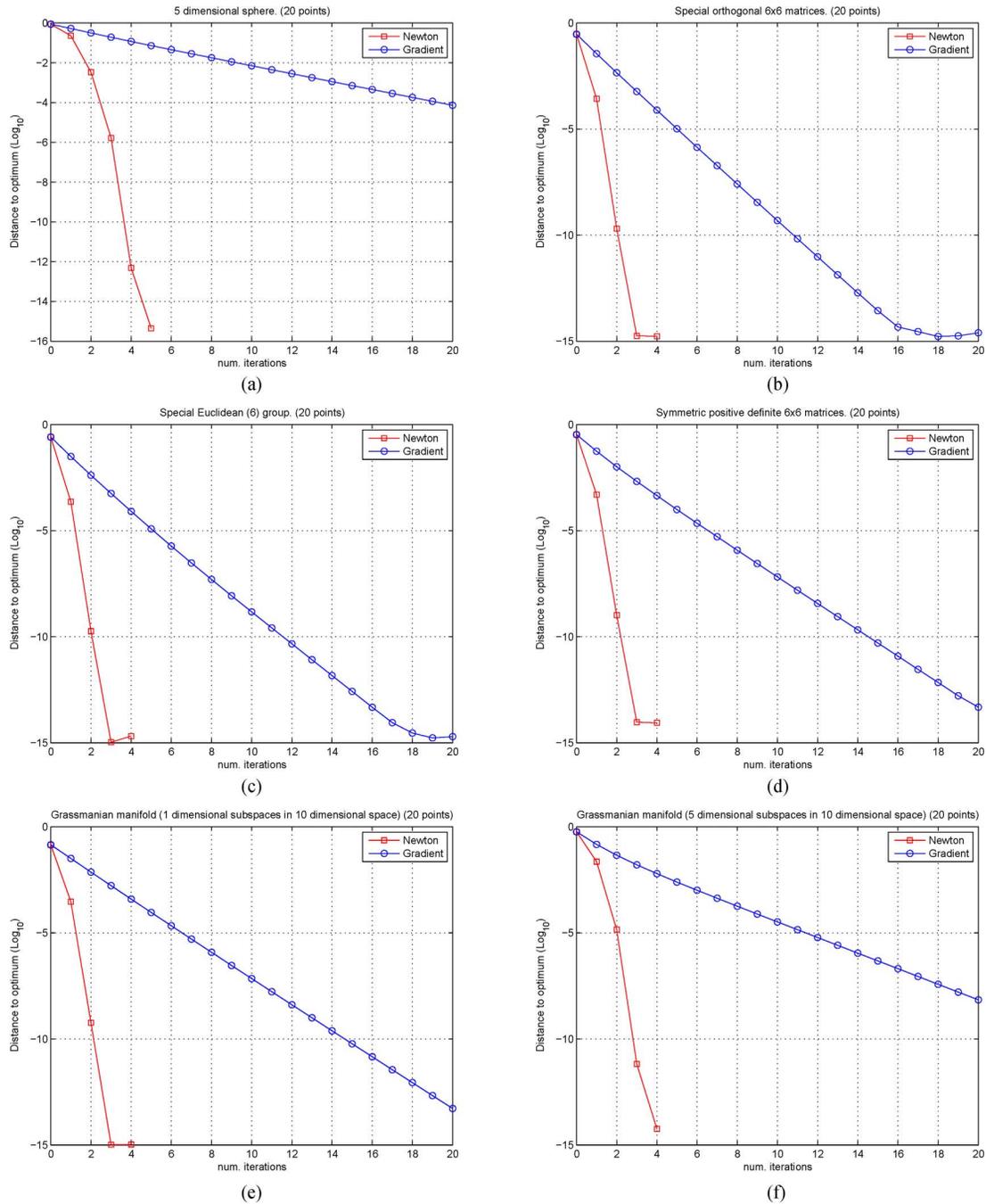


Fig. 1. Simulation results for centroid computation. Except for the Grassman manifold, whose constellations were built with a radius of $\pi/6$, all constellations were built with a radius of $\pi/3$.

A. NRHS Construction for a Particular Riemannian Manifold

When faced with an optimization problem on a particular Riemannian manifold M , it is not usually known whether or not it admits an NRHS structure. Since many useful manifolds in engineering admit such structures, the process of identifying it will be described here.

First it is necessary to describe M as a coset manifold $M \cong G/H$ where the symbol \cong states that the two sides are diffeomorphic. Here, a proposition stated in [9] solves the problem, stating that all that needs to be done is to find a Lie group G which acts transitively on M :

Theorem A.1: Let $G \times M \rightarrow M$ be a transitive action and let H be its isotropy subgroup at a point $o \in M$. Then there is a natural map $F : G/H \rightarrow M$ which is a diffeomorphism. In particular, the projection $\hat{F} : G \rightarrow M, g \mapsto go$ is a submersion.

Furthermore, this action must be an isometry as stated in the definition of an NRHS space, which means that for any $p \in M, X_p, Y_p \in T_p M$ and $g \in G$ the property $\langle X_p, Y_p \rangle_p = \langle g_* X_p, g_* Y_p \rangle_{gp}$, must hold, where g_* denotes the push forward of the translation by g .

Examples:

- 1) $\mathbb{S}\mathbb{O}(n+1) \subset \mathbb{G}\mathbb{L}(n+1)$ acts on the unit sphere $\mathbb{S}^n \subset \mathbb{R}(n+1)$ (seen as a Riemannian subspace) as the restriction

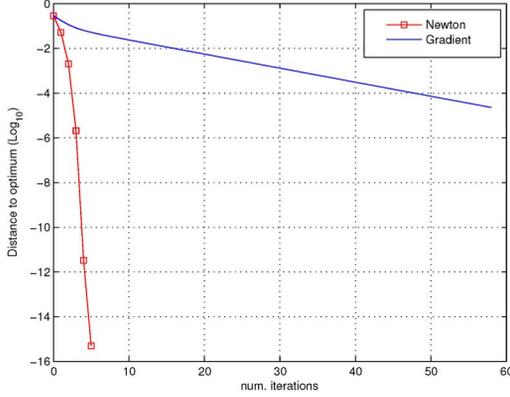


Fig. 2. MAP position estimate results using 5 observed landmarks with $\sigma = 1$ and $R = I$.

of the usual action of $\mathbb{G}\mathbb{L}(n+1)$ on $\mathbb{R}(n+1)$. This action is transitive. The isotropy subgroup of $o = (1, 0, \dots, 0) \in \mathbb{S}^n$ consists of the subgroup

$$H = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & q \end{bmatrix} \in \mathbb{S}\mathbb{O}(n+1) : q \in \mathbb{S}\mathbb{O}(n) \right\} \cong \mathbb{S}\mathbb{O}(n)$$

Hence, ignoring the natural diffeomorphism yields $\mathbb{S}^n \cong \mathbb{S}\mathbb{O}(n+1)/\mathbb{S}\mathbb{O}(n)$. To verify that this action is G-invariant let $X_p, Y_p \in T_p\mathbb{S}^n$ and $g \in \mathbb{S}\mathbb{O}(n+1)$. Then:

$$\langle g_*X_p, g_*Y_p \rangle_{gp} = \text{tr} \{ X_p^T g^T g Y_p \} = \text{tr} \{ X_p^T Y_p \} = \langle X_p, Y_p \rangle_p.$$

- 2) As a trivial example, $\mathbb{S}\mathbb{O}(n)$ acts transitively on itself (seen as a Riemannian submanifold of $\mathbb{G}\mathbb{L}(n+1)$ with the Euclidean inner product) through group multiplication. The isotropy subgroup at any point is the trivial subgroup $H = \{e\}$ (where e is the group identity), hence trivially $\mathbb{S}\mathbb{O}(n) \cong \mathbb{S}\mathbb{O}(n)/\{e\}$. As before, to verify that this action preserves the inner product let $X_p, Y_p \in T_p\mathbb{S}\mathbb{O}(n)$ and $g \in \mathbb{S}\mathbb{O}(n)$. Then:

$$\langle g_*X_p, g_*Y_p \rangle_{gp} = \text{tr} \{ X_p^T g^T g Y_p \} = \langle X_p, Y_p \rangle_p.$$

- 3) Expanding the previous example, the Lie group product $G = \mathbb{S}\mathbb{O}(n) \times \mathbb{R}^n$ acts transitively on the Special Euclidean group $M = \mathbb{S}\mathbb{E}(n)$ (seen as a Riemannian submanifold of $\mathbb{G}\mathbb{L}(n+1)$) as

$$\begin{aligned} (\mathbb{S}\mathbb{O}(n) \times \mathbb{R}^n) \times \mathbb{S}\mathbb{E}(n) &\longrightarrow \mathbb{S}\mathbb{E}(n) \\ \left((R, v), \begin{bmatrix} Q & t \\ 0 & 1 \end{bmatrix} \right) &\longmapsto \begin{bmatrix} RQ & t+v \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Once again the isotropy subgroup is trivial, hence $\mathbb{S}\mathbb{E}(n) \cong \mathbb{S}\mathbb{O}(n) \times \mathbb{R}^n$. If $X_p = \begin{bmatrix} K_X & \Delta_X \\ 0 & 0 \end{bmatrix} \in T_pM$ then for a given element $g = (R_g, \Delta_g) \in G$ the push forward of the action of g is given by:

$$g_*X_p = g_* \begin{bmatrix} K_X & \Delta_X \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_g K_X & t_X \\ 0 & 0 \end{bmatrix}$$

Hence the action preserves the inner product since:

$$\begin{aligned} \langle g_*X_p, g_*Y_p \rangle_{gp} &= \text{tr} \left\{ \begin{bmatrix} R_g K_X & \Delta_X \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} R_g K_Y & \Delta_Y \\ 0 & 0 \end{bmatrix} \right\} \\ &= \text{tr} \left\{ \begin{bmatrix} K_X^T K_Y & - \\ - & \Delta_X^T \Delta_Y \end{bmatrix} \right\} \\ &= \langle X_p, Y_p \rangle_p. \end{aligned}$$

- 4) $\mathbb{G}\mathbb{L}(n)$ (the set of $n \times n$ invertible matrices with real entries) acts transitively on $\text{Sym}^+(n)$ (the set of $n \times n$ positive definite symmetric matrices with the inner product described below) by conjugation, that is $(g, p) \mapsto g^T p g$. The isotropy subgroup of the identity matrix seen as an element of $\text{Sym}^+(n)$ is the set $H = \{g : g^T g = e\} = \mathbb{O}(n)$. So, $\text{Sym}^+(n) \cong \mathbb{G}\mathbb{L}(n)/\mathbb{O}(n)$. Letting $X_p, Y_p \in T_p\text{Sym}^+(n)$ and $g \in \mathbb{G}\mathbb{L}(n)$ and assuming the inner product is given by $\langle X_p, Y_p \rangle_p = \text{tr} \{ X_p^T p^{-1} Y_p p^{-1} \}$, then:

$$\begin{aligned} \langle g_*X_p, g_*Y_p \rangle_{gp} &= \text{tr} \{ g^T X_p^T g g^{-1} p^{-1} g^{-T} g^T Y_p g g^{-1} p^{-1} g^{-T} \} \\ &= \text{tr} \{ X_p^T p^{-1} Y_p p^{-1} \} = \langle X_p, Y_p \rangle_p \end{aligned}$$

- 5) If $S \in \mathcal{M}(n, r)$ (the set of $n \times r$ real matrices) with $n > r$ and S is full rank, let the notation $[S]$ denote the subspace generated by the column vectors of S . The Grassman manifold consists of the set of all such subspaces, i.e. $\mathbb{G}(n, r) = \{[S] : S \in \mathcal{M}(n, r), n > r, \text{rank}(S) = r\}$. Please note that the elements of the Grassman manifold are equivalence classes, where $[S_1] = [S_2] \Leftrightarrow$ the columns of S_1 and the columns of S_2 span the same subspace. Consider the transitive action of $G = \mathbb{S}\mathbb{O}(n)$ on $M = \mathbb{G}(n, r)$ defined as

$$\mathbb{S}\mathbb{O}(n) \times \mathbb{G}(n, r) \longrightarrow \mathbb{G}(n, r), \quad (g, [S]) \longmapsto [gS]$$

the isotropy subgroup of $o = \begin{bmatrix} I \\ 0 \end{bmatrix} \in \mathbb{G}(n, r)$ is the set

$$H = \left\{ \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} : (Q_1, Q_2) \in S(\mathbb{O}r \times \mathbb{O}n-r) \right\}$$

hence $\mathbb{G}(n, r) \cong \mathbb{S}\mathbb{O}(n)/S(\mathbb{O}(r) \times \mathbb{O}(n-r))$. The action is G-invariant since once again if $X_p, Y_p \in T_p\mathbb{G}(n, r)$ and $g \in \mathbb{S}\mathbb{O}(n)$ then:

$$\langle g_*X_p, g_*Y_p \rangle_{gp} = \text{tr} \{ X_p^T g^T g Y_p \} = \text{tr} \{ X_p^T Y_p \} = \langle X_p, Y_p \rangle_p.$$

Let \mathfrak{g} be the Lie algebra of G and \mathfrak{h} be the Lie sub-algebra of H . The next step consists of finding a Lie subspace \mathfrak{m} such that $\mathfrak{m} + \mathfrak{h} = \mathfrak{g}$ and $\text{Ad}_H \mathfrak{m} = \mathfrak{m}$. This step must be done by inspection but it is usually not hard to accomplish.

Examples:

- 1) Let $\text{Skew}(n)$ denote the set of $n \times n$ skew symmetric matrices with real entries. For the coset space $\mathbb{S}^n = \mathbb{S}\mathbb{O}(n+1)/\mathbb{S}\mathbb{O}(n)$, $\mathfrak{g} = \text{Skew}(n+1)$ and $\mathfrak{h} = \left\{ \begin{bmatrix} 0 & 0 \\ 0 & k \end{bmatrix} : k \in \text{Skew}(n) \right\}$. By inspection (due to the requirement that $\mathfrak{m} + \mathfrak{h} = \mathfrak{g}$) a logical candidate for \mathfrak{m} is the set $\mathfrak{m} = \left\{ \begin{bmatrix} 0 & -x^T \\ x & 0 \end{bmatrix} : x \in \mathbb{R}^n \right\}$. Since
- $$\begin{aligned} \text{Ad}_H \mathfrak{m} &= \left\{ \begin{bmatrix} 1 & 0 \\ 0 & q \end{bmatrix} \begin{bmatrix} 0 & -x^T \\ x & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & q^T \end{bmatrix} \right\} \\ &= \left\{ \begin{bmatrix} 0 & -(qx)^T \\ qx & 0 \end{bmatrix} : x \in \mathbb{R}^n, q \in \mathbb{S}\mathbb{O}(n) \right\} = \mathfrak{m} \end{aligned} \quad (5)$$

results that \mathfrak{m} is indeed a Lie subspace.

- 2) When $G = \mathbb{S}\mathbb{O}(n)$ and $H = \{e\}$, $\mathfrak{g} = Skew(n)$ and $\mathfrak{h} = 0$ (the trivial vector space). Hence, the obvious choice is $\mathfrak{m} = \mathfrak{g}$, which is obviously invariant under $Ad_H(\cdot)$.
- 3) The same happens when considering $G = \mathbb{S}\mathbb{O}(n) \times \mathbb{R}^n$ and $H = \{(e, 0)\}$. Hence, for $\mathbb{S}\mathbb{E}(n)$, $\mathfrak{m} = \mathfrak{g} = Skew(n) \times \mathbb{R}^n$.
- 4) If $G = \mathbb{G}\mathbb{L}(n)$ and $H = \mathbb{O}(n)$ as is the case for $M = Sym^+(n)$, the corresponding Lie algebras are $\mathfrak{g} = \mathcal{M}(n)$ ($n \times n$ real matrices) and $\mathfrak{h} = Skew(n)$. A natural candidate for the Lie subspace is the set of symmetric matrices $\mathfrak{m} = Sym(n)$, and indeed if $q \in \mathbb{O}(n)$ and $X_e \in Sym(n)$ then $qX_eq^T = qX_e^Tq^T = (qX_eq^T)^T$ hence it is $Ad_H(\cdot)$ invariant.
- 5) For the case of the Grassmann, $G = \mathbb{S}\mathbb{O}(n)$ and $H = S(\mathbb{O}(r) \times \mathbb{O}(n-r))$ (as seen previously). The corresponding Lie algebras are $\mathfrak{g} = Skew(n)$ and $\mathfrak{h} = \left\{ \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} : (K_1, K_2) \in Skew(r) \times Skew(n-r) \right\}$. Then, by inspection, the obvious choice for the Lie subspace is

$$\mathfrak{m} = \left\{ \begin{bmatrix} 0 & M \\ -M^T & 0 \end{bmatrix} : M \in \mathbb{R}^{r \times n-r} \right\}.$$

It is easily checked that this choice is $Ad_H(\cdot)$ invariant.

All that remains to be done is to verify if the construction verifies the property

$$\langle [X, Y]_{\mathfrak{m}}, Z \rangle = \langle X, [Y, Z]_{\mathfrak{m}} \rangle, \quad \text{for } X, Y, Z \in \mathfrak{m}$$

Since \mathfrak{m} is identified with T_eM , the dot product is the pull-back by \hat{F} of the dot product on M . If the property is not satisfied, another construction with another Lie group G acting on M might be tried, or it is possible that M does not admit an NRHS structure.

Examples:

- 1) Continuing the previous examples consider the sphere, where $X_e, Y_e, Z_e \in \mathfrak{m} \subset Skew(n+1)$. Let $X_e = \begin{bmatrix} 0 & -x^T \\ x & 0 \end{bmatrix}$ and $Y_e = \begin{bmatrix} 0 & -y^T \\ y & 0 \end{bmatrix}$ where $x, y \in \mathbb{R}^n$. Then it results that $[X_e, Y_e]_{\mathfrak{m}} = (X_e Y_e - Y_e X_e)|_{\mathfrak{m}} = \begin{bmatrix} 0 & 0 \\ 0 & yx^T - xy^T \end{bmatrix}|_{\mathfrak{m}} = 0$. Thus the required result is trivially verified. If needed, the corresponding dot product on \mathfrak{m} can be found by noting that $F_* X_e = \begin{bmatrix} 0 \\ x \end{bmatrix}$. Hence
- 2) For $M = G = \mathbb{S}\mathbb{O}(n)$, the tangent vectors in both manifolds are canonically identified, hence if $X_e, Y_e, Z_e \in \mathfrak{m}$ the inner product on \mathfrak{m} is given in the usual way:

$$\langle X_e, Y_e \rangle_{\mathfrak{m}} = \text{tr} \{ X_e^T Y_e \}$$

So the required property is once again satisfied

$$\begin{aligned} \langle [X_e, Y_e]_{\mathfrak{m}}, Z \rangle &= \text{tr} \{ (X_e Y_e - Y_e X_e)^T Z_e \} \\ &= \text{tr} \{ Y_e X_e Z_e - X_e Y_e Z_e \} \\ &= \text{tr} \{ -X_e^T Z_e Y_e + X_e^T Y_e Z_e \} \\ &= \langle X_e, [Y_e, Z_e]_{\mathfrak{m}} \rangle \end{aligned}$$

- 3) When $M = \mathbb{S}\mathbb{E}(n)$ and $G = \mathbb{S}\mathbb{O}(n) \times \mathbb{R}^n$, the inner product on \mathfrak{m} is found in the same manner as in the first example. Thus if $X_e = (K_X, \Delta_X)$, $Y_e = (K_Y, \Delta_Y)$, $Z_e = (K_Z, \Delta_Z) \in \mathfrak{m}$

$$\begin{aligned} \langle X_e, Y_e \rangle_{\mathfrak{m}} &= \langle \hat{F}_* X_e, \hat{F}_* Y_e \rangle_M \\ &= \text{tr} \left\{ \begin{bmatrix} K_X & \Delta_X \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} K_Y & \Delta_Y \\ 0 & 0 \end{bmatrix} \right\} \\ &= \text{tr} \{ K_X^T K_Y \} + \Delta_X^T \Delta_Y \\ &= \langle K_X, K_Y \rangle_{\mathbb{S}\mathbb{O}(n)} + \langle \Delta_X, \Delta_Y \rangle_{\mathbb{R}^n} \end{aligned}$$

The Lie bracket on the product group is given by the product of the Lie brackets. Hence

$$\begin{aligned} [X_e, Y_e]_{\mathbb{S}\mathbb{O}(n) \times \mathbb{R}^n} &= \left([K_X, K_Y]_{\mathbb{S}\mathbb{O}(n)}, [\Delta_X, \Delta_Y]_{\mathbb{R}^n} \right) \\ &= (K_X K_Y - K_Y K_X, 0) \end{aligned}$$

Then to check the required property:

$$\begin{aligned} \langle [X_e, Y_e]_{\mathfrak{m}}, Z \rangle &= \text{tr} \{ (K_X K_Y - K_Y K_X)^T K_Z \} + 0 \\ &= \langle X_e, [Y_e, Z_e]_{\mathfrak{m}} \rangle \end{aligned}$$

(check the example for the $\mathbb{S}\mathbb{O}(n)$ case for details of the last step).

- 4) In the case of the symmetric positive definite matrices where $\mathfrak{m} = Sym(n)$ the Lie bracket results in a skew symmetric matrix, hence the projection back to \mathfrak{m} results in a null vector. Hence the requirement is trivially satisfied.
- 5) Noting that for the Grassmann manifold $X_e, Y_e \in \mathfrak{m} \subset Skew(n)$ then $[X_e, Y_e]_{\mathfrak{m}} = (X_e Y_e - Y_e X_e)|_{\mathfrak{m}} = 0$ The needed property is once again trivially verified.

Hence all five manifolds have been described as naturally reductive homogeneous spaces.

B. Operations on NRHS Manifolds

This section details why the structure of these manifolds is important to the Riemannian optimization process. A proposition in [9] states:

Theorem A.2: If $M = G/H$ is a naturally reductive homogeneous space, its geodesic starting at o with tangent vector $X_o \in T_oM$ are given by $\gamma(t) = \hat{F} \circ \alpha(t)$ for all $t \in \mathbb{R}$, where $\alpha(t)$ is the one parameter subgroup of X_o identified as an element of \mathfrak{m} .

Hence the Riemannian exponential map follows directly from the Lie group's exponential map which in our examples is the standard matrix exponential (since G is either $\mathbb{G}\mathbb{L}(n)$ or $\mathbb{S}\mathbb{O}(n)$). Geodesics starting at any other point of M can be found by translation of γ since G acts transitively as an isometry. The Riemannian logarithm map follows from inversion.

On a manifold with NRHS structure, the curvature endomorphism is also computable as seen for example in [26]:

$$\begin{aligned} R(X_p, Y_p) \cdot Z_p &= \left[Z_p, [X_p, Y_p]_{\mathfrak{h}} \right] + \frac{1}{2} [Z_p, [X_p, Y_p]_{\mathfrak{m}}]_{\mathfrak{m}} \\ &\quad + \frac{1}{4} [[X_p, Z_p]_{\mathfrak{m}}, Y_p]_{\mathfrak{m}} + \frac{1}{4} [X_p, [Y_p, Z_p]_{\mathfrak{m}}]_{\mathfrak{m}} \quad (6) \end{aligned}$$

Examples: To finish the examples, we provide a final summary of the functions needed for each of the considered manifolds. Note that these are simplified and concise versions of the results obtained by the above theorem and expression 6.

1) *The Sphere* \mathbb{S}^n : This n -dimensional manifold is described as the set $\mathbb{S}^n = \{x \in \mathbb{R}(n+1) : \|x\| = 1\}$ whose tangent space at a point $p \in \mathbb{S}^n$ is $T_p\mathbb{S}^n \cong \{x \in \mathbb{R}(n+1) : p^T x = 0\}$. Let $p, q \in \mathbb{S}^n$, $X_p, Y_p, Z_p \in T_p\mathbb{S}^n$ and s is the norm of X_p . It can be shown that for the ambient metric $\langle X_p, Y_p \rangle = X_p^T Y_p$:

- $\exp_p(X_p) = p \cos(s) + (X_p/s) \sin(s)$.
- $\log_p(q) = (q - p(p^T q))(a/\sin(a))$ where $a = \arccos(p^T q)$.
- $R(X_p, Y_p) \cdot Z_p = \langle Y_p, Z_p \rangle X_p - \langle X_p, Z_p \rangle Y_p$.

2) *Special Orthogonal Group* $\mathbb{SO}(n)$: This $n(n-1)/2$ dimensional manifold represents the set of rotations of \mathbb{R}^n and is described as $\mathbb{SO}(n) = \{x \in \mathcal{M}(n, n) : x^T x = \mathbf{I}\}$ whose tangent space at a point $p \in \mathbb{SO}(n)$ is $T_p\mathbb{SO}(n) \cong \{pk : k \in \text{Skew}(n)\}$, where $\text{Skew}(n)$ denotes the set of $n \times n$ skew-symmetric matrices. The metric comes naturally from the Riemannian embedding as $\langle X_p, Y_p \rangle = \text{tr}\{X_p^T Y_p\}$

- $\exp_p(X_p) = p \exp(p^T X_p)$, where \exp denotes the matrix exponential function.
- $\log_p(q) = p \log(p^T q)$ where \log denotes the matrix logarithm function.
- $R(X_p, Y_p) \cdot Z_p = -(1/4)[[X_p, Y_p], Z_p]$.

3) *Special Euclidean Group* $\mathbb{SE}(n)$: The Euclidean group, characterized as the product manifold $\mathbb{SO}(n) \times \mathbb{R}^n$, inherits these manifolds' properties. Hence, at a point $p = (R, t)$, the tangent space $T_p\mathbb{SE}(n) = T_R\mathbb{SO}(n) \times T_t\mathbb{R}^n$ with dot product given by $\langle (X_R, v_t), (Y_R, u_t) \rangle_{\mathbb{SE}(n)} = \langle X_R, Y_R \rangle_{\mathbb{SO}(n)} + \langle v_t, u_t \rangle_{\mathbb{R}^n}$.

- $\exp_{(R,t)}((X, v)_{(R,t)}) = (R \exp(R^T X_R), t + v_t)$, where \exp denotes the matrix exponential function.
- $\log_{(R,t)}((Q, s)) = (R \log(R^T Q), s - t)$, where \log denotes the matrix logarithm function.
- $R((X, v)_{(R,t)}, (Y, u)_{(R,t)}) \cdot (Z, w)_{(R,t)} = (-1/4)[[X_R, Y_R], Z_R], 0$ where the brackets denote the Lie bracket only on $\mathfrak{so}(n)$ since \mathbb{R}^n is flat.

4) *Symmetric Positive Definite Matrices* $\text{Sym}^+(n)$: This $n(n+1)/2$ dimensional manifold is described as the set $\text{Sym}^+(n) = \{x \in \mathcal{M}(n, n) : x = x^T, \text{with positive eigenvalues}\}$ whose tangent space at a point $p \in \text{Sym}^+(n)$ is $T_p\text{Sym}^+(n) \cong \{x : x \in \text{Sym}(n)\}$, where $\text{Sym}(n)$ denotes the set of $n \times n$ symmetric matrices. Let $p, q \in \text{Sym}^+(n)$, $X_p, Y_p, Z_p \in T_p\text{Sym}^+(n)$. When considering the metric $\langle X_p, Y_p \rangle = \text{tr}\{X_p^T p^{-1} Y_p p^{-1}\}$ the following expressions hold

- $\exp_p(X_p) = p^{1/2} \exp(p^{-1/2} X_p p^{-1/2}) p^{1/2}$.
- $\log_p(q) = p^{1/2} \log(p^{-1/2} q p^{-1/2}) p^{1/2}$.
- $R(X_p, Y_p) \cdot Z_p = 1/4(Z_p p^{-1} O - O p^{-1} Z_p)$, where $O = X_p p^{-1} Y_p - Y_p p^{-1} X_p$.

5) *The Grassmann Manifold* $\mathbb{G}(n, r)$: The Grassmann is an $r(n-r)$ dimensional manifold of r dimensional linear subspaces in \mathbb{R}^n . It is naturally described as a quotient manifold with the previously mentioned equivalence relation, and a point $p = [P]$ is described by a representative $P \in \mathbb{O}(n, r)$ which is the set of the first r columns of the elements in $\mathbb{O}(n)$ for

$n > r$. The tangent space at a point $p = [P] \in \mathbb{G}(n, r)$ is $T_p\mathbb{G}(n, r) \cong \{x \in \mathcal{M}(n \times r) : P^T x = 0\}$.

Let $p = [P], q = [Q] \in \mathbb{G}(n, r)$, $X_p, Y_p, Z_p \in T_p\mathbb{G}(n, r)$. For the ambient metric $\langle X_p, Y_p \rangle = \text{tr}\{X_p^T Y_p\}$ (the simplified expressions shown next are obtained from [2], [27] and the curvature endomorphism follows directly from equation 6):

- $\exp_p(X_p) = [[PVU]_{\begin{smallmatrix} \cos \Sigma \\ \sin \Sigma \end{smallmatrix}} V^T]$, where $X_p = U\Sigma V^T$ is the compact SVD of X_p .
- $\log_p(q) = (-PV\Sigma + QV)DU^T$, where $U\Sigma V^T$ is the SVD decomposition of $P^T Q$, σ_i are the diagonal elements of Σ and $D = \text{diag}([\cos^{-1}(\sigma_1)/\sin(\cos^{-1}(\sigma_1))] \dots [\cos^{-1}(\sigma_r)/\sin(\cos^{-1}(\sigma_r))])$.
- $R(X_p, Y_p) \cdot Z_p = [Z_p, [X_p, Y_p]]$.

APPENDIX B

PROOF OF THE THEOREMS

This section proves theorem IV.1 and corollary IV.2, a main contribution of this paper but delayed to an appendix for readability. The proof is mostly a chain of known results which can be found in texts such as [9], [28]. With the intent of finding the Hessian of the function k_p , recall that (see for example [9])

$$\text{Hess}k_p(q)(X_q, Y_q) = \left\langle (\nabla_X \text{grad}k_p)_q, Y_q \right\rangle \quad (7)$$

where X is any local extension of X_q . Note that from the properties of a connection, its value depends only of X at q , but for the expression to be formally correct the extension X must be considered. Knowing that the gradient operator is linear and that for any two smooth functions $f, g : U \rightarrow \mathbb{R}$ defined on an open set $U \subset M$ satisfies the point-wise multiplication property $\text{grad}(fg) = f \text{grad}g + g \text{grad}f$, allows for the simplification $\text{grad}k_p = (1/2)\text{grad}(r_p r_p) = r_p \text{grad}r_p$.

Defining $\partial/\partial r_p$ as the unit normed radial vector field when written in normal coordinates centered at p , a corollary to Gauss's Lemma [25] states that $\text{grad}r_p = \partial/\partial r_p$. Hence the former expression is written as $\text{grad}k_p = r_p(\partial/\partial r_p)$. Gauss's Lemma also allows for the decomposition of any vector field $X \in U$ as $X = X^\perp + X^\parallel$, where $X^\parallel \in U$ is a vector field parallel to $\partial/\partial r_p$ and $X^\perp \in U$ is orthogonal to it. These statements, along with the properties of a connection, are used to write

$$\begin{aligned} \nabla_X \text{grad}k_p &= \nabla_X \left(r_p \frac{\partial}{\partial r_p} \right) \\ &= X(r_p) \frac{\partial}{\partial r_p} + r_p \nabla_{(X^\parallel + X^\perp)} \left(\frac{\partial}{\partial r_p} \right) \\ &= X(r_p) \frac{\partial}{\partial r_p} + r_p \nabla_{X^\parallel} \left(\frac{\partial}{\partial r_p} \right) \\ &\quad + r_p \nabla_{X^\perp} \left(\frac{\partial}{\partial r_p} \right). \end{aligned}$$

Noting that for any vector field X , $X(r_p) = dr_p(X) = \langle \text{grad}r_p, X \rangle = \langle (\partial/\partial r_p), X \rangle$ and since X^\parallel is parallel to $\partial/\partial r_p$, there is a smooth function $f : U \rightarrow \mathbb{R}$ such that

$X^\parallel = f(\partial/\partial r_p)$. Since $\partial/\partial r_p$ is tangent to unit speed geodesics emanating from p , $\nabla_{\partial/\partial r_p} \partial/\partial r_p = 0$. Hence

$$\begin{aligned} \nabla_X \text{grad} k_p &= \left\langle \frac{\partial}{\partial r_p}, X \right\rangle \frac{\partial}{\partial r_p} + r_p f \nabla_{\frac{\partial}{\partial r_p}} \left(\frac{\partial}{\partial r_p} \right) \\ &\quad + r_p \nabla_{X^\perp} \left(\frac{\partial}{\partial r_p} \right) \\ &= X^\parallel + r_p \nabla_{X^\perp} \left(\frac{\partial}{\partial r_p} \right) \end{aligned} \quad (8)$$

Now let $\theta : \mathbb{R} \rightarrow S_r \subset M$ be a curve in the geodesic sphere $S_r = \{s \in M : r_p(s) = r = r_p(q)\}$ with $\theta(0) = q$ and $\dot{\theta}(0) = X_q^\perp$. In normal coordinates centered at p , consider the geodesic variation $\hat{\Gamma} :]-\delta, \delta[\times [0, r]$ for some δ , given by $\hat{\Gamma}(s, t) = (t/r)\hat{\theta}(s)$. Here the hat notation denotes a coordinate representation, hence if $\phi : M \rightarrow \mathbb{R}^n$ is the normal coordinate function then $\hat{p} = \phi(p)$, $\hat{\theta} = \phi \circ \theta$, and $\hat{r}_p = r_p \circ \phi^{-1}$.

Defining $\hat{T}(s, t) = (\partial \hat{\Gamma} / \partial t)(s, t)$ and $\hat{S}(s, t) = (\partial \hat{\Gamma} / \partial s)(s, t)$, the corresponding Jacobi field $J : M \rightarrow TM$ (see [25]) is given in coordinates by $\hat{J}(t) = \hat{S}(0, t) = (t/r)\hat{X}_q^\perp$. Note that \hat{J} is normal to the unit-speed geodesic $\hat{\gamma}(t) = \hat{\Gamma}(0, t) = (t/r)\hat{q}$ and that $\hat{J}(r) = \hat{X}_q^\perp$. Also, notice that $\hat{T}(s, t) = \phi_*(\partial/\partial r_p)|_{\hat{\Gamma}(s, t)}$ since $\hat{\Gamma}(0, t)$ is a geodesic. In order to ease notation the coordinate representation for these objects will be hidden, although not forgotten. Hence, at a point q , $\nabla_{X^\perp}(\partial/\partial r_p)|_q = D_s T(0, r) = D_t S(0, r) = D_t J(r)$ where the fact that $D_s T = D_t S$ is used (see [25] Lemma 6.3). Substituting in (8), again at q , results in

$$(\nabla_X \text{grad} k_p)|_q = X_q^\parallel + r \nabla_{X^\perp} \left(\frac{\partial}{\partial r_p} \right) \Big|_q = X_q^\parallel + r D_t J(r)$$

Substituting back into (7), yields the known expression for the Hessian in terms of Jacobi fields [12], [23]

$$\begin{aligned} \text{Hess} k_p(q)(X_q, Y_q) &= \langle X_q^\parallel + r D_t J(r), Y_q \rangle \\ &= \langle X_q^\parallel, Y_q \rangle + r \langle D_t J(r), Y_q \rangle \end{aligned} \quad (9)$$

All that remains to do is to find an expression for the Jacobi field and take its covariant derivative. This leads to a rather lengthy discussion so it is stated here as a couple of lemmas:

Lemma B.1: The solution of the ODE $\ddot{u} + ku = 0$, $u(0) = a$, $\dot{u}(0) = b$ is given by $u(t) = ac_k(t) + bs_k(t)$ where

$$c_k(t) = \begin{cases} \cos(\sqrt{k}t) & k > 0 \\ 1 & k = 0 \\ \cosh(\sqrt{-k}t) & k < 0 \end{cases},$$

$$s_k(t) = \begin{cases} \frac{1}{\sqrt{k}} \sin(\sqrt{k}t) & k > 0 \\ t & k = 0 \\ \frac{1}{\sqrt{-k}} \sinh(\sqrt{-k}t) & k < 0 \end{cases}$$

Proof: Picard's existence theorem guarantees uniqueness and direct substitution of the result in the differential equation proves the result. \square

Lemma B.2: Let M be a locally-symmetric Riemannian manifold and $B(p)$ be a geodesic ball centered at $p \in M$. If $r = r_p(q)$ is the intrinsic distance of a point $q \in B(p)$ to p and $\gamma : [0, r] \rightarrow B(p)$ is the unit speed radial geodesic running from p to q , given a tangent vector $V_q \in T_q M$ orthogonal to

$\dot{\gamma}(r)$, then the normal Jacobi field $J : [0, r] \rightarrow TM$ along γ which satisfies $J(0) = 0$ and $J(r) = V_q$ is given by

$$J(t) = \sum_{i=1}^n \left[\frac{\langle V_q, E_i(r) \rangle}{s_{\lambda_i}(r)} s_{\lambda_i}(t) \right] E_i(t) \quad (10)$$

where $E_i(t) \in TM$ is the parallel transport along γ of the tangent space's orthonormal basis $\{E_{i_q}\} \subset T_q M$ which diagonalizes the linear operator $\mathcal{R} : T_q M \rightarrow T_q M$ defined as $\mathcal{R}(X_q) = R(X_q, \dot{\gamma}(0)) \cdot \dot{\gamma}(0)$, i.e., $\mathcal{R}(E_{i_q}) = \lambda_i E_{i_q}$. Note that R denotes the curvature tensor of M and s_{λ_i} is defined as in the previous lemma.

Proof: As stated in [25] the Jacobi field J specified by its two endpoints $J(0) = 0$ and $J(r) = V_q$ exists and is unique if q is not conjugate to p along γ . Consider the Jacobi equation

$$\ddot{J}(t) + R(J(t), \dot{\gamma}(t)) \cdot \dot{\gamma}(t) = 0 \quad (11)$$

Choosing an orthonormal basis for the tangent space $\{F_{i_q}\} \subset T_q M$ and creating the vector fields $F_i \in TM$ along γ by parallel translation of F_{i_q} , one can write $J(t)$ with respect to this basis (note that the set $\{F_{i_{\gamma(t)}}\} \subset T_{\gamma(t)} M$ is a basis for $T_{\gamma(t)} M$) as $J(t) = \sum_{i=1}^n J^i(t) F_i(t)$, where $J^i : [0, r] \rightarrow \mathbb{R}$. Hence the left hand side of (11) can be written as

$$\begin{aligned} &\ddot{J}(t) + R(J(t), \dot{\gamma}(t)) \cdot \dot{\gamma}(t) \\ &= \sum_{i=1}^n \ddot{J}^i(t) F_i(t) + R\left(\sum_{i=1}^n J^i(t) F_i(t), \dot{\gamma}(t)\right) \cdot \dot{\gamma}(t) \\ &= \sum_{i=1}^n \ddot{J}^i(t) F_i(t) + J^i(t) R(F_i(t), \dot{\gamma}(t)) \cdot \dot{\gamma}(t) \end{aligned} \quad (12)$$

where the identity $\ddot{J}(t) = \sum_{i=1}^n \ddot{J}^i(t) F_i(t)$ is used, which follows from the fact that the vector fields $F_i(t)$ are parallel. The goal is now to solve this ordinary differential equation. Start by defining the linear operator $\mathcal{R} : T_q M \rightarrow T_q M$ as $\mathcal{R}(X_q) = R(X_q, \dot{\gamma}(0)) \cdot \dot{\gamma}(0)$ and note that due to the symmetries of the Riemannian curvature tensor, this operator is self-adjoint, i.e. $\langle \mathcal{R}(X_q), Y_q \rangle = \langle X_q, \mathcal{R}(Y_q) \rangle$. This guarantees that there is an orthonormal basis $E_{i_q} \in T_q M$ such that $\mathcal{R}(E_{i_q}) = \lambda_i E_{i_q}$ for some $\lambda_i \in \mathbb{R}$ as described next. Write $\mathcal{R}(F_{i_q})$ as a linear combination as $\mathcal{R}(F_{i_q}) = \sum_{j=1}^n \langle F_{j_q}, \mathcal{R}(F_{i_q}) \rangle F_{j_q}$. Since any two vector spaces with the same dimension are isomorphic there is an isomorphism, $\phi : T_p M \rightarrow \mathbb{R}^n$ taking F_{i_q} to $\hat{F}_i \in \mathbb{R}^n$. In this vector space the operator \mathcal{R} may be written as $\hat{\mathcal{R}} = \phi \circ \mathcal{R} \circ \phi^{-1}$. Hence, writing in matrix notation by defining $\hat{F} = [\hat{F}_1 \ \hat{F}_2 \ \dots \ \hat{F}_n]$ and since \mathcal{R} is linear in a finite dimensional vector field, $\hat{\mathcal{R}}$ can be described as a matrix. Hence,

$$\hat{\mathcal{R}} \hat{F} = \underbrace{\begin{bmatrix} \langle F_{1_q}, \mathcal{R}(F_{1_q}) \rangle & \langle F_{1_q}, \mathcal{R}(F_{2_q}) \rangle & \dots & \langle F_{1_q}, \mathcal{R}(F_{n_q}) \rangle \\ \langle F_{2_q}, \mathcal{R}(F_{1_q}) \rangle & \langle F_{2_q}, \mathcal{R}(F_{2_q}) \rangle & \dots & \langle F_{2_q}, \mathcal{R}(F_{n_q}) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle F_{n_q}, \mathcal{R}(F_{1_q}) \rangle & \langle F_{n_q}, \mathcal{R}(F_{2_q}) \rangle & \dots & \langle F_{n_q}, \mathcal{R}(F_{n_q}) \rangle \end{bmatrix}}_A \hat{F}$$

The fact that the operator is self-adjoint makes A a symmetric matrix and as such, it admits an eigenvalue decomposition $A = UDU^T$ where $D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ is a diagonal matrix

such that λ_i are the eigenvalues of A and U is an orthogonal matrix with the normalized eigenvectors of A as its columns. Hence $\hat{\mathcal{R}}\hat{F} = \hat{F}UDU^T \iff \hat{\mathcal{R}}\hat{F}U = \hat{F}UD$ hence, if $U = [u_{ij}]$ are the entries of the matrix, $\hat{E}_j = \sum_{i=1}^n u_{ij}\hat{F}_i$ is the orthonormal basis such that $\hat{\mathcal{R}}\hat{E}_j = \lambda_j\hat{E}_j$. Using the isomorphism once again, this means that $E_{j_q} = \sum_{i=1}^n u_{ij}F_{i_q}$ is the basis that diagonalizes the operator \mathcal{R} as $\mathcal{R}(E_{i_q}) = \lambda_i E_{i_q}$. Define $E_i(t)$ as the parallel transport of E_{i_q} along γ . Define as well the vector fields $R_i(t) = R(E_i(t), \dot{\gamma}(t)) \cdot \dot{\gamma}(t)$ which, since M is locally symmetric, are parallel along $\gamma(t)$ [9]. It follows then, using the fact that parallel transport preserves inner products and $\{E_{i_q}\}$ is an orthonormal basis

$$\begin{aligned} R_i(t) &= \sum_{k=1}^n \langle R_i(t), E_k(t) \rangle E_k(t) = \sum_{k=1}^n \langle R_i(r), E_k(r) \rangle E_k(t) \\ &= \sum_{k=1}^n \langle \mathcal{R}(E_{i_q}), E_{k_q} \rangle E_k(t) = \lambda_i \sum_{k=1}^n \langle E_{i_q}, E_{k_q} \rangle E_k(t) \\ &= \lambda_i E_i(t) \end{aligned}$$

Writing (12) in terms of the new basis:

$$\begin{aligned} \ddot{J}(t) + R(J(t), \dot{\gamma}(t)) \cdot \dot{\gamma}(t) &= \sum_{i=1}^n \ddot{J}^i(t) E_i(t) + J^i(t) \underbrace{R(E_i(t), \dot{\gamma}(t)) \cdot \dot{\gamma}(t)}_{R_i(t)} \\ &= \sum_{i=1}^n \left(\ddot{J}^i(t) + \lambda_i J^i(t) \right) E_i(t) \end{aligned}$$

Hence, the Jacobi equation decouples into n scalar equations with the corresponding two-point boundary conditions:

$$\begin{cases} \ddot{J}^1(t) + \lambda_1 J^1(t) = 0, & J^1(0) = 0 \text{ and } J^1(r) = \langle V_q, E_1(r) \rangle \\ \ddot{J}^2(t) + \lambda_1 J^2(t) = 0, & J^2(0) = 0 \text{ and } J^2(r) = \langle V_q, E_2(r) \rangle \\ \vdots & \\ \ddot{J}^n(t) + \lambda_1 J^n(t) = 0, & J^n(0) = 0 \text{ and } J^n(r) = \langle V_q, E_n(r) \rangle \end{cases}$$

Invoking lemma B.1, the solution of the i th equation is given by $J^i(t) = (\langle V_q, E_i(r) \rangle / s_{\lambda_i}(r)) s_{\lambda_i}(t)$. Hence $J(t) = \sum_{i=1}^n [(\langle V_q, E_i(r) \rangle / s_{\lambda_i}(r)) s_{\lambda_i}(t)] E_i(t)$. \square

This expression for $J(t)$ can be substituted in (9), by making $V_q = X_q^\perp$. Taking the covariant derivative of $J(t)$ evaluated at $t = r$, considering that $D_t E_i(t) = 0$ since $E_i(t)$ is parallel along the geodesic, results in

$$D_t J(r) = \sum_{i=1}^n \left[\frac{\langle X_q^\perp, E_i(r) \rangle}{s_{\lambda_i}(r)} c_{\lambda_i}(r) \right] E_i(r)$$

Hence, defining by pointwise division $\text{ctg}_{\lambda_i}(r) = r(c_{\lambda_i}(r)/s_{\lambda_i}(r))$,

$$\begin{aligned} \text{Hess}k_p(q)(X_q, Y_q) &= \langle X_q^\parallel, Y_q \rangle + r \langle D_t J(r), Y_q \rangle \\ &= \langle X_q^\parallel, Y_q \rangle \\ &\quad + \sum_{i=1}^n \text{ctg}_{\lambda_i}(r) \langle X_q^\perp, E_i(r) \rangle \langle Y_q, E_i(r) \rangle \end{aligned}$$

Although (2) provides a way to compute the Hessian, it is not very implementation-friendly. This section proves corollary IV.2 which re-writes the equation in matrix form once a tangent basis $\{F_{i_q}\} \subset T_q M$ is fixed.

From the symmetries of the curvature tensor follows that any vector parallel to $\dot{\gamma}(r)$ (for example X_q^\parallel) belongs to the kernel of the operator \mathcal{R} . Without loss of generality assume that $E_1(r)$ is parallel to $\dot{\gamma}(r)$ (hence $\lambda_1 = 0$, $X_q^\parallel = \langle X_q, E_1(r) \rangle E_1(r)$ and $X_q^\perp = \sum_{j=2}^n \langle X_q, E_j(r) \rangle E_j(r)$). Then, since $\text{ctg}_0(r) = 1$, (2) can be re-written:

$$\begin{aligned} \text{Hess}k_p(q)(X_q, Y_q) &= \text{ctg}_{\lambda_1}(r) \left\langle \langle X_q, E_1(r) \rangle E_1(r), \sum_{j=1}^n \langle Y_q, E_j(r) \rangle E_j(r) \right\rangle \\ &\quad + \sum_{i=1}^n \text{ctg}_{\lambda_i}(r) \left\langle \sum_{j=2}^n \langle X_q, E_j(r) \rangle E_j(r), E_i(r) \right\rangle \\ &\quad \times \langle Y_q, E_i(r) \rangle \\ &= \text{ctg}_{\lambda_1}(r) \langle X_q, E_1(r) \rangle \langle Y_q, E_1(r) \rangle \\ &\quad + \sum_{i=2}^n \text{ctg}_{\lambda_i}(r) \langle X_q, E_i(r) \rangle \langle Y_q, E_i(r) \rangle \\ &= \sum_{i=1}^n \text{ctg}_{\lambda_i}(r) \langle X_q, E_i(r) \rangle \langle Y_q, E_i(r) \rangle = \hat{X}^T H_{k_p} \hat{Y} \end{aligned}$$

where the matrices are defined in the theorem statement.

REFERENCES

- [1] D. Gabay, "Minimizing a differentiable function over a differential manifold," *J. Optim. Theory Appl.*, vol. 37, no. 2, pp. 177–219, 1982.
- [2] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353, 1998.
- [3] J. H. Manton, "Optimisation algorithms exploiting unitary constraints," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 635–650, Mar. 2002.
- [4] C. Belta and V. Kumar, "An svd-based projection method for interpolation on SE(3)," *IEEE Trans. Robot. Autom.*, vol. 18, no. 3, pp. 334–345, Jun. 2002.
- [5] J. H. Manton, "A centroid (Karcher mean) approach to the joint approximate diagonalization problem: The real symmetric case," *Digital Signal Process.*, vol. 5, pp. 468–478, 2005.
- [6] U. Helmke, K. Hüper, P. Lee, and J. Moore, "Essential matrix estimation via Newton-type methods," in *Proc. 16th Int. Symp. Math. Theory Netw. Syst. (MTNS)*, Leuven, Belgium, 2004.
- [7] K. Krakowski, K. Hüper, and J. Manton, "On the computation of the Karcher mean on spheres and special orthogonal groups," in *Proc. Workshop Robot. Math. (RoboMat '07)*.
- [8] P. Absil, R. Mahony, and R. Sepulchre, "Riemannian geometry of grassmann manifolds with a view on algorithmic computation," *Acta Appl. Math.*, vol. 80, no. 2, pp. 199–220, Jan. 2004.
- [9] B. O'Neil, *Semi-Riemannian Geometry*. New York, NY, USA: Academic, 1983.
- [10] A. Arvanitoyeorgos, *An Introduction to Lie Groups and the Geometry of Homogeneous Spaces*. Providence, RI, USA: Amer. Math. Soc., 1999.
- [11] M. Moakher, "Means and averaging in the group of rotations," *SIAM J. Matrix Anal. Appl.*, vol. 24, no. 1, pp. 1–16, 2002.
- [12] J. H. Manton, "A globally convergent numerical algorithm for computing the centre of mass on compact Lie groups," in *Proc. 8th Int. Conf. Control, Autom., Robot., Vis.*, Kunming, China, Dec. 2004.

- [13] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Res. Rep. 5255, INRIA, Jul. 2004, Int. J. Comput. Vis.*.
- [14] P. Fletcher, C. Lu, S. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Trans. Med. Imag.*, vol. 23, no. 8, pp. 995–1005, Aug. 2004.
- [15] D. Groisser, "Newton's method, zeroes of vector fields, and the Riemannian center of mass," *Adv. Appl. Math.*, vol. 33, pp. 95–135, 2004.
- [16] R. Ferreira, J. Xavier, J. Costeira, and V. Barroso, "Newton method for Riemannian centroid computation in naturally reductive homogeneous spaces," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP'06)*, May 2006, pp. 704–707.
- [17] R. Ferreira and J. Xavier, "Hessian of the Riemannian squared distance function on connected locally symmetric spaces with applications," in *Proc. Control '06*.
- [18] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*. New York, NY, USA: Academic, 1975.
- [19] M. P. do Carmo, *Riemannian Geometry*. Boston, MA, USA: Birkhäuser, 1992.
- [20] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.
- [21] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.
- [22] K. Hüper and J. Trumppf, "Newton-like methods for numerical optimization on manifolds," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2004, 2006, pp. 136–139.
- [23] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Comm. Pure Appl. Math.*, vol. 30, pp. 509–541, 1977.
- [24] B. Afsari, "Riemannian Lp center of mass: Existence, uniqueness, and convexity," *Proc. Amer. Math. Soc.*, vol. 139, pp. 655–673, 2011.
- [25] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*. New York, NY, USA: Springer, 1997.
- [26] I. Agricola, "Connections on naturally reductive spaces, their dirac operator and homogeneous models in string theory," *Commun. Math. Phys.*, no. 232, pp. 535–563, 2003.
- [27] D. Sepiashvili, J. M. F. Moura, and V. H. S. Ha, "Affine-permutation symmetry: Invariance and shape space," in *Proc. IEEE Workshop Statist. Signal Process.*, Sep.–1 Oct. 2003, pp. 307–310.
- [28] J. Cheeger and D. G. Ebin, *American Mathematical Society, Comparison Theorems in Riemannian Geometry*. Providence, RI, USA: AMS Chelsea Publishing, 1975.



Ricardo Ferreira graduated in Electrical and Computer Engineering in 2004 at Instituto Superior Técnico (IST). In 2006 he received his M.Sc. degree in studying underwater stereo reconstructions of 3D scenes when observed through an air-water interface. His Ph.D. studies focused on reconstructing paperlike surfaces from multiple camera images, and were concluded in 2010. Both his M.Sc. and Ph.D. degrees were obtained at IST. His current research interests focus on general geometric problems arising in robotics and computer vision.



João Xavier received the PhD degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 2002. Currently, he is an assistant professor in the Department of Electrical and Computer Engineering, IST. He is also a researcher at the Institute of Systems and Robotics (ISR), Lisbon, Portugal. His current research interests are in the area of optimization, sensor networks, and signal processing on manifolds. He is a member of the IEEE.



João P. Costeira received the Licenciatura degree, MS degree, and PhD degree in electrical and computer engineering from Instituto Superior Técnico (IST), Lisbon, Portugal in 1985, 1989, and 1995, respectively. He was a visiting scientist at the Robotics Institute, Carnegie Mellon University between 1992 and 1995, where he developed work in the area of multiple motion segmentation. Currently he is associate professor at IST and a researcher at Instituto de Sistemas e Robótica working in the area of computer vision.



Victor Barroso (M'89–SM'00) was born in Lisbon, Portugal, on December 22, 1952. He received the E.E. degree in 1976 and the Ph.D. degree in electrical and computer engineering in 1990, both from Instituto Superior Técnico (IST), Lisbon, Portugal. He received the title of "Professor com Agregação" in Electrical and Computer Engineering in 2003 from Universidade Técnica de Lisboa-IST. In 1976, he joined the faculty of the Department of Electrical and Computer Engineering, IST. He was promoted to Assistant Professor in 1990, to Associate Professor in 1993, and to Full Professor in 2004. He has taught courses in systems and signal theory, control systems, signal and array processing, computer networks, and communications. He was Vice-President of the IST's Scientific Council from November 2001 to March 2007. From April 2007 to March 2009, he was the Director of the Portuguese pole of the Information and Communication Technology Institute (ICTI@Portugal), a partnership involving several Portuguese Universities, research and development institutions, and companies with the Carnegie Mellon University. He was also the Director of the Institute for Systems and Robotics (ISR), Lisbon, Portugal, from September 2009 to December 2012. His research interests are in statistical signal/image and array processing and wireless communications. He has published over 100 journal and conference papers.

Dr. Barroso was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and for the IEEE SIGNAL PROCESSING LETTERS. He was the representative of the IEEE Signal Processing Society at the IEEE Wireless Periodicals Working Group in 2000 and the General Chair of the IEEE International Workshop on Signal Processing Advances in Wireless Communications in 2004. He is affiliated with IEEE Signal Processing Society.