

MULTIVARIATE HERMITE INTERPOLATION OF MANIFOLD-VALUED DATA

Ralf Zimmermann*

Ronny Bergmann[†]

2022-12-15

In this paper, we propose two methods for multivariate Hermite interpolation of manifold-valued functions. On the one hand, we approach the problem via computing suitable weighted Riemannian barycenters. To satisfy the conditions for Hermite interpolation, the sampled derivative information is converted into a condition on the derivatives of the associated weight functions. It turns out that this requires the solution of linear systems of equations, but no vector transport is necessary. This approach treats all given sample data points equally and is intrinsic in the sense that it does not depend on local coordinates or embeddings.

As an alternative, we consider Hermite interpolation in a tangent space. This is a straightforward approach, where one designated point, for example one of the sample points or (one of) their center(s) of mass, is chosen to act as the base point at which the tangent space is attached. The remaining sampled locations and sampled derivatives are mapped to said tangent space. This requires a vector transport between different tangent spaces. The actual interpolation is then conducted via classical vector space operations. The interpolant depends on the selected base point.

The validity and performance of both approaches is illustrated by means of numerical examples.

Keywords. Hermite interpolation, algorithms on manifolds, Riemannian center of mass, barycentric interpolation, Karcher mean

AMS subject classifications (MSC2010). 65D05 65D15 49Q99 41A29 53B50

*Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark (zimmermann@imadasdu.dk, <https://portal.findresearcher.sdu.dk/en/persons/zimmermann> orcid: 0000-0003-1692-3996);

[†]Norwegian University of Science and Technology, Department of Mathematical Sciences, Trondheim, Norway (ronny.bergmann@ntnu.no, <https://www.ntnu.edu/employees/ronny.bergmann>, orcid: 0000-0001-8342-7218).

1. INTRODUCTION

In this paper, we address multivariate Hermite interpolation of a function that takes values on a complete Riemannian manifold \mathcal{M} with tangent bundle $T\mathcal{M}$. More precisely, let $D \subset \mathbb{R}^d$ be a parameter domain and consider a differentiable function

$$f: D \rightarrow \mathcal{M}, \quad \omega \mapsto f(\omega).$$

Consider a sample data set consisting of k parameter locations $\omega_1, \dots, \omega_k \in D$ with corresponding function values (manifold locations) and partial derivatives (tangent vectors)

$$p_j = f(\omega_j) \in \mathcal{M}, \quad v_j^i := \partial_i f(\omega_j) = \left. \frac{d}{dt} \right|_{t=0} f(\omega_j + te_i) = df(\omega_j)[e_i] \in T_{f(\omega_j)}\mathcal{M},$$

where e_1, \dots, e_d denotes the orthonormal basis of unit vectors in \mathbb{R}^d . The Hermite manifold interpolation problem is formalized as follows:

Find a differentiable, manifold-valued function $\hat{f}: D \rightarrow \mathcal{M}$ such that

$$\hat{f}(\omega_j) = p_j \in \mathcal{M}, \quad j = 1, \dots, k \tag{1a}$$

$$\partial_i \hat{f}(\omega_j) = v_j^i \in T_{p_j}\mathcal{M}, \quad j = 1, \dots, k; \quad i = 1, \dots, d. \tag{1b}$$

1.1. ORIGINAL CONTRIBUTIONS

We develop two approaches to tackle the multivariate manifold Hermite interpolation problem (1a), (1b). The first one is via computing weighted Riemannian barycenters. This can be considered as an extension of the interpolation approach developed in Grohs, Hardering, and Sander 2015 and in Sander 2012, 2016 by incorporating derivative data. We refer to this as barycentric Hermite interpolation (**BHI**). The BHI method has the following main features:

- (i) The approach works on arbitrary Riemannian manifolds, i.e., no special structure (Lie Group, homogeneous space, symmetric space,...) is required. In order to conduct practical computations, only an algorithm for evaluating the Riemannian logarithm map must be available. All occurrences of Riemannian exponentials can be replaced with retractions and analogously any Riemannian logarithms by inverse retractions Absil, Mahony, and Sepulchre 2008, Section 4.1. For clarity of presentation we just write exponential and logarithmic map, respectively, during this paper.
- (ii) The differentiability order of the interpolant is the same as that of the weight functions. Hence, when working with smooth weight functions, a smooth interpolant is produced.
- (iii) Although our theoretical derivation involves covariant derivatives and the Hessian forms of scalar fields on manifolds, the practical implementation of BHI does not require computing any such operators.
- (iv) The method works only locally, on a domain, where the Riemannian center of mass exists and is unique. The number of sample points must exceed the dimension of the manifold.

The second approach is a straightforward translation of Hermite interpolation in Euclidean vector spaces to a selected tangent space of the manifold under consideration. We refer to this approach as tangent space Hermite interpolation (**THI**). The THI method has the following main features:

- (i) The approach works on arbitrary Riemannian manifolds, i.e., no special structure (Lie Group, homogeneous space, symmetric space,...) is required. In order to conduct practical computations, algorithms for evaluating the Riemannian exponential map and the Riemannian logarithm map must be available (or must be consistently replaced with invertible retractions). The differential of the Riemannian logarithm must be evaluated (or approximated).
- (ii) The differentiability order of the interpolant is the same as that of the weight functions.
- (iii) The method works only locally, on a star-shaped domain around a manifold location $p_c \in \mathcal{M}$, where the Riemannian normal coordinates are well-defined. The interpolant is constructed in the tangent space $T_{p_c}\mathcal{M}$ and thus depends on the choice of the center point p_c .

We illustrate the validity and the performance of both approaches by means of numerical examples.

1.2. RELATED WORK

In the research literature, there are two different tracks of manifold interpolation research. On the one hand, there is the problem of interpolating scalar functions with manifold-valued *inputs* $f: \mathcal{M} \rightarrow \mathbb{R}$. This line of research is followed, e.g., in Allasia, Roberto, and De Rossi 2018; Narcowich 1995, but is not considered here. On the other hand, interpolating parametric functions with manifold-valued *outputs* $f: \mathbb{R}^d \rightarrow \mathcal{M}$ is investigated. The work at hand subordinates to this setting. Confusion may be caused by the fact that for both types of problems, one speaks of “interpolation on a manifold”. To the best of our knowledge, *multivariate* Hermite interpolation for output data on general manifolds has not yet been considered in the research literature. Univariate Hermite interpolation has been considered explicitly in Jakubiak, Leite, and Rodrigues 2006 for data on compact, connected Lie groups with a bi-invariant metric. A general approach to Hermite curve interpolation is featured in Zimmermann 2020.

A related line of research is the generalization of Bézier curves and the De Casteljaeu-algorithm to Riemannian manifolds, see Bergmann and Gousenbourger 2018; Gousenbourger, Massart, and Absil 2018; Nava-Yazdani and Polthier 2013; Popiel and Noakes 2007; Samir and Adouani 2019. The transition of this technique to manifolds is via replacing the inherent straight lines with geodesics. The start and end velocities of the resulting spline are proportional to the velocity vectors of the geodesics that connect the first two and the last two control points, respectively (Popiel and Noakes 2007, Theorem 1). Hence, the method can be adapted to work for univariate Hermite interpolation. More general subdivision schemes based on similar geodesic averages have been investigated in Dyn and Sharon 2017; Wallner and Dyn 2005. The preprint Vardi, Dyn, and Sharon 2022 addresses univariate Hermite interpolation on the sphere via subdivision schemes. Two-variate manifold interpolation via Bézier surfaces with C^1 -connection of local patches has been investigated in Absil, Gousenbourger, et al. 2016.

A multivariate Hermite-type method that is specifically tailored to interpolation problems on the Grassmann manifold is sketched in Amsallem 2010, §3.7.4.

Interpolation with Riemannian barycenters has been proposed in the context of geodesic finite elements in Grohs, Hardering, and Sander 2015 and in Sander 2012, 2016.

An overview over manifold interpolation methods with a special focus on model reduction applications is given in Zimmermann 2021.

1.3. NOTATIONAL CONVENTIONS

Throughout, we assume that \mathcal{M} is a complete Riemannian manifold of dimension $\dim(\mathcal{M}) = m$. The tangent space of \mathcal{M} at $p \in \mathcal{M}$ is $T_p\mathcal{M}$ and the tangent bundle is $T\mathcal{M} = \cup_{p \in \mathcal{M}} T_p\mathcal{M}$. As a Riemannian manifold, \mathcal{M} carries a family of inner products $\{\langle \cdot, \cdot \rangle_p : T_p\mathcal{M} \times T_p\mathcal{M} \rightarrow \mathbb{R}, (v, w) \mapsto \langle v, w \rangle_p\}_{p \in \mathcal{M}}$ that depends smoothly on the base point $p \in \mathcal{M}$.

The differential of a function $f: \mathbb{R}^d \rightarrow \mathbb{R}^m$ at a point p is denoted by $Df(p): \mathbb{R}^d \rightarrow \mathbb{R}^m$. For a two-arguments function $L: \mathbb{R}^m \times \mathbb{R}^d \rightarrow \mathbb{R}^m$, $(q, \omega) \mapsto L(q, \omega)$, the linear maps $D_q L(q, \omega): \mathbb{R}^m \rightarrow \mathbb{R}^m$ and $D_\omega L(q, \omega): \mathbb{R}^d \rightarrow \mathbb{R}^m$ denote the differentials of $q \mapsto L(q, \omega)$ (with ω considered as fixed) and $\omega \mapsto L(q, \omega)$ (with q considered as fixed), respectively. The differential at a point p of a function $f: \mathcal{M} \rightarrow \mathcal{N}$ between differentiable manifolds \mathcal{M}, \mathcal{N} is denoted by $df(p): T_p\mathcal{M} \rightarrow T_{f(p)}\mathcal{N}$. For $L: \mathcal{M} \times \mathbb{R}^d \rightarrow \mathcal{M}$, $(q, \omega) \mapsto L(q, \omega)$, the notations $d_q L(q, \omega)$ and $d_\omega L(q, \omega)$ are to be understood in analogy to the above.

A vector field is a smooth function $X: \mathcal{M} \rightarrow T\mathcal{M}$ such that $X(p) \in T_p\mathcal{M}$ for all $p \in \mathcal{M}$. The set of smooth vector fields on \mathcal{M} is denoted by $\mathcal{X}(\mathcal{M})$. We use the symbol ∇ to denote the (unique) Levi-Civita connection on \mathcal{M} . For $X, Y \in \mathcal{X}(\mathcal{M})$, the notation $\nabla_X Y \in \mathcal{X}(\mathcal{M})$ denotes the covariant derivative of the vector field Y in the direction of the vector field X . A vector field along a curve $c: I \rightarrow \mathcal{M}$ is a mapping $X_c: I \rightarrow T\mathcal{M}$ such that $X_c(t) \in T_{c(t)}\mathcal{M}$. The covariant derivative of X_c along c is $\frac{D}{dt} X_c$. If there is an ambient vector field $X \in \mathcal{X}(\mathcal{M})$ such that $X(c(t)) = X_c(t)$, then $\frac{D}{dt} X_c = \nabla_{\dot{c}(t)} X$. For a scalar function $g: \mathcal{M} \rightarrow \mathbb{R}$, the gradient vector field $\text{grad } g \in \mathcal{X}(\mathcal{M})$ is point-wise defined by $\langle \text{grad } g(p), v \rangle_p = dg(p)[v]$ for all $v \in T_p\mathcal{M}$. The covariant derivative of the gradient field yields the Hessian $\text{Hess } g[X] = \nabla_X \text{grad } g$ and gives rise to the endomorphism $T_p\mathcal{M} \ni v \mapsto \text{Hess } g(p)[v] = (\nabla_v \text{grad } g)(p) \in T_p\mathcal{M}$.

For a scalar two-parameter function $L: \mathcal{M} \times \mathbb{R}^d, (q, \omega) \mapsto L(q, \omega) \in \mathbb{R}$, we will write $\text{grad}_q L(p, \omega)$ for the gradient by q evaluated at (p, ω) , which is defined by $\langle \text{grad}_q L(p, \omega), v \rangle_p = d_q L(p, \omega)[v]$, for all $v \in T_p\mathcal{M}$. Likewise, $\text{Hess}_q L(p, \omega)[v]$ denotes the Hessian of $L(q, \omega)$ by q evaluated at (p, ω) applied to the tangent vector $v \in T_p\mathcal{M}$.

1.4. ORGANIZATION OF THE PAPER

Section 2 recaps the barycentric interpolation methods. In Section 3, the approach is extended to the Hermite setting. The alternative approach of tangent space Hermite interpolation is outlined in Section 4. Numerical experiments are featured in Section 5, and Section 6 concludes the paper.

2. INTERPOLATION WITH WEIGHTED RIEMANNIAN BARYCENTERS

The Riemannian barycenter or Riemannian center of mass¹ or Fréchet mean of a sample data set $\{p_1, \dots, p_k\} \subset \mathcal{M}$ on a manifold is defined as the minimizer of the Riemannian objective function

$$\mathcal{M} \ni q \mapsto L(q) = \frac{1}{2} \sum_{j=1}^k w_j \operatorname{dist}(q, p_j)^2,$$

where $\operatorname{dist}(q, p_j)$ is the Riemannian distance between the manifold locations q and p_j and $w_j \geq 0$ are scalar weights such that $\sum_{j=1}^k w_j = 1$. Formally, the latter requirements mean that the center of mass is taken with respect to a discrete positive measure of unit weight. This definition generalizes the notion of the barycentric mean in Euclidean spaces, cf. [Appendix A](#) of the supplements. However, on curved manifolds, the global center might not be unique. Moreover, local minimizers may appear. For more details, see [Karcher 1977](#) and [Afsari, Tron, and Vidal 2013](#), which also give uniqueness criteria.

2.1. INTERPOLATION VIA OPTIMIZATION

Interpolation can be performed by computing weighted Riemannian centers. More precisely, let $f: \mathbb{R}^d \supset D \rightarrow \mathcal{M}$ and let $\omega_1, \dots, \omega_k \subset D$ be a set of parameter locations and let $p_j = f(\omega_j) \in \mathcal{M}$, $j = 1, \dots, k$ be the corresponding sampled manifold locations on \mathcal{M} . The interpolant is then defined on the convex hull $\operatorname{conv}\{\omega_1, \dots, \omega_k\} \subset D \subset \mathbb{R}^d$ of the samples.

Let $\{\varphi_j : \omega \mapsto \varphi_j(\omega) \in \mathbb{R} \mid j = 1, \dots, k\}$ be a suitable set of multivariate, scalar-valued interpolation weight functions with $\varphi_l(\omega_j) = \delta_{lj}$ and $\sum_{j=1}^k \varphi_j(\omega) \equiv 1$. Such weight functions can be constructed, e.g., as Lagrangians, [Sander 2016](#), or radial basis functions, [Buhmann 2003](#). The interpolant $q^* \approx f(\omega^*) \in \mathcal{M}$ at an unsampled parameter location $\omega^* \in \operatorname{conv}\{\omega_1, \dots, \omega_k\}$ can be taken to be the minimizer

$$\omega^* \mapsto f(\omega^*) = q^* := \arg \min_{q \in \mathcal{M}} L(q, \omega^*), \quad \text{where } L(q, \omega) := \frac{1}{2} \sum_{j=1}^k \varphi_j(\omega) \operatorname{dist}(q, p_j)^2. \quad (2)$$

Since the weight functions may attain negative values, this corresponds to taking the center of mass with respect to a discrete *signed* measure of unit weight, see the discussion in [Sander 2016](#), Section 3. Computing q^* thus requires one to solve a Riemannian optimization problem. At sample location ω_l , one has indeed that

$$2L(q, \omega_l) = \sum_{j=1}^k \varphi_j(\omega_l) \operatorname{dist}(q, p_j)^2 = \sum_{j=1}^k \delta_{lj} \operatorname{dist}(q, p_j)^2 = \operatorname{dist}(q, p_l)^2,$$

which has the unique global minimum at $q^* = p_l$. Hence, the function $\hat{f}: \omega \mapsto \arg \min_{q \in \mathcal{M}} L(q, \omega)$ satisfies the basic interpolation conditions.

¹Here, we introduce Riemannian barycenter for discrete data sets; for centers w.r.t. a general mass distribution, see [Karcher's original paper Karcher 1977](#), Section 1.

Remark 2.1. *Under certain conditions that locally ensure the existence and uniqueness of Riemannian barycenters, the minima of (2) are exactly at the zeros of the associated gradient. Moreover, they depend smoothly on the parameters (q, ω) if the weight functions φ_j are smooth, see Sander 2016, Theorems 3.19 & 4.1. As a consequence, the interpolant is smooth under these conditions.*

2.2. THE GRADIENT AND HESSIAN OF THE RIEMANNIAN DISTANCE FUNCTION

Interpolation via barycenters is an optimization task. As a rule, numerical optimization requires the computation of the gradient of the objective function. The next theorem, due to Karcher, provides the gradient of the squared Riemannian distance function.

Theorem 2.2 (Karcher 1977, Thm. 1.2). *Let \mathcal{M} be a complete Riemannian manifold and let $p \in \mathcal{M}$. Let $B_\rho(p)$ be a geodesic ball of radius ρ around p such that the geodesics between any two points inside $B_\rho(p)$ are unique and minimizing. Define*

$$L_p: B_\rho(p) \rightarrow \mathbb{R}, \quad q \mapsto \frac{1}{2} \text{dist}(q, p)^2.$$

Then

$$\text{grad } L_p(q) = -\text{Log}_q(p),$$

where $\text{Log}_q = (\text{Exp}_q)^{-1}$ is the Riemannian logarithm map.

Because Karcher works with barycenters with respect to mass distributions and also to make this exposition self-contained, we recap the proof in [Appendix B](#).

Eventually, we also need information on the Hessian of $L_p(q) = \frac{1}{2} \text{dist}(q, p)^2$ at $q = p$. The Riemannian Hesse form of a scalar function $g: \mathcal{M} \rightarrow \mathbb{R}$ at q is

$$\text{Hess } g(q): T_q\mathcal{M} \rightarrow T_q\mathcal{M}, \quad v \mapsto \text{Hess } g(q)[v] = (\nabla_v \text{grad } g)(q), \quad (3)$$

see Kühnel 2015, §6A.

Remark 2.3. *Actually, the Riemannian Hesse $(1, 1)$ -tensor maps vector fields to vector fields,*

$$\text{Hess } g: \mathcal{X}(\mathcal{M}) \rightarrow \mathcal{X}(\mathcal{M}), \quad X \mapsto \nabla_X \text{grad } g.$$

Yet, since a tensor is a point-wise object, $(\nabla_v \text{grad } g)(q) = (\nabla_X \text{grad } g)(q)$ for all vector fields $X \in \mathcal{X}(\mathcal{M})$ with $X(q) = v$. Hence, it makes sense to consider the Hessian at q as an endomorphism of $T_q\mathcal{M}$. Additional background information on the Riemannian Hessian is given in [Appendix B](#) of the supplements.

Theorem 2.4. *Consider the setting of Theorem 2.2. For p fixed, the Hesse form of the function $q \mapsto L_p(q) = \frac{1}{2} \text{dist}(q, p)^2$ at p is the identity,*

$$\text{Hess } L_p(p) = \text{id}_{T_p\mathcal{M}}: T_p\mathcal{M} \rightarrow T_p\mathcal{M}.$$

Proof. We use Karcher's approach of computing the Hessian via a variation through geodesics, see [Theorem B.3](#) and [Figure 8](#) in the supplement. This shows that for a geodesic $t \mapsto \gamma(t)$ with $\gamma(0) = q$, $\dot{\gamma}(0) = v$ and the associated variation of geodesics

$$(s, t) \mapsto c_p(s, t) = \text{Exp}_p(s \text{Log}_p(\gamma(t))),$$

it holds

$$\langle \text{Hess } L_p(q)[v], v \rangle = \left\langle \frac{D}{ds} \partial_t c_p(1, 0), v \right\rangle.$$

In the special case, where the geodesic γ starts from $q = p$ with velocity $v \in T_p \mathcal{M}$, we obtain

$$\begin{aligned} \partial_t|_{t=0} c_p(s, t) &= d \left(\text{Exp}_p \right)_{s \text{Log}_p(\gamma(0))} \left(\text{sd} \left(\text{Log}_p \right)_{\gamma(0)} (\dot{\gamma}(0)) \right) \\ &= d \left(\text{Exp}_p \right)_{s \cdot 0} \left(\text{sd} \left(\text{Log}_p \right)_p (v) \right) = sv \in T_p \mathcal{M}, \end{aligned}$$

because $d(\text{Exp}_p)_0 = \text{id}_{T_p \mathcal{M}} = d(\text{Log}_p)_p$. Note that $s \mapsto \partial_t c_p(s, 0) \in T_p \mathcal{M}$ is a vector field along the point curve $\alpha: s \mapsto \alpha(s) \equiv p$. Therefore, the covariant derivative coincides with the usual derivative and we obtain

$$\frac{D}{ds} \partial_t c_p(s, 0) = \frac{d}{ds} (sv) = v.$$

As a consequence,

$$\langle \text{Hess } L_p(p)[v], v \rangle = \langle v, v \rangle \quad \text{for all } v \in T_p \mathcal{M}.$$

The Hessian is symmetric. Via polarization, it is uniquely determined by terms of the above form. This yields $\text{Hess } L_p(p)[v] = v$. \square

By [Theorem 2.2](#), the gradient of the objective function L in (2) by q is

$$\text{grad}_q L(q, \omega) = \sum_{j=1}^k \varphi_j(\omega) \cdot (-\text{Log}_q(p_j)) \in T_q \mathcal{M}. \quad (4)$$

At a sample location p_l , the gradient (by q) of the objective function $L(q, \omega)$ in the barycentric interpolation problem (2) vanishes, because of $\text{Log}_{p_l}(p_l) = 0 \in T_{p_l} \mathcal{M}$. By [Theorem 2.4](#), the Hessian (again with respect to the q -argument) is

$$\text{Hess}_q L(p_l, \omega_l)[v] = \sum_{j=1}^k \varphi_j(\omega_l) \text{Hess } L_{p_j}(p_l)[v] = \text{Hess } L_{p_l}(p_l)[v] = v \in T_{p_l} \mathcal{M}, \quad (5)$$

because $\varphi_j(\omega_l) = \delta_{lj}$. Thus, the Hessian by q at (p_l, ω_l) is the identity on $T_{p_l} \mathcal{M}$. In particular, it has full rank. A generic gradient descent algorithm² to compute the barycentric interpolant for a function $f: \mathbb{R}^d \ni \omega \mapsto f(\omega) \in \mathcal{M}$ is given in [Algorithm 1](#)

Algorithm 1. *Interpolation via the weighted Riemannian barycenter.*

²There exist several different ways of computing the weighted Riemannian barycenter, see for example <https://juliamanifolds.github.io/Manifolds.jl/latest/features/statistics.html> for an overview.

Input: Sample data set $\{p_1 = f(\omega_1), \dots, p_k = f(\omega_k)\} \subset \mathcal{M}$, unsampled parameter location $\omega^* \in \text{conv}(\omega_1, \dots, \omega_k) \subset \mathbb{R}^d$, initial guess q_0 , convergence threshold τ

- 1: $k := 0$
- 2: Compute $\text{grad}_q L(q_k, \omega^*)$ according to (4)
- 3: **while** $\|\text{grad}_q L(q_k, \omega^*)\|_q > \tau$ **do**
- 4: select a step size α_k
- 5: $q_{k+1} := \text{Exp}_{q_k}^{\mathcal{M}}(-\alpha_k \text{grad}_q L(q_k, \omega^*))$
- 6: $k := k + 1$
- 7: **end while**

Output: $\hat{f}(\omega^*) := q^* := q_k \in \mathcal{M}$ interpolant of $f(\omega^*)$.

3. BARYCENTRIC HERMITE INTERPOLATION

In this section, we enhance the method of weighted barycentric interpolation by including derivative information. The task is to construct an interpolant of the form

$$\hat{f}: \omega \mapsto \hat{f}(\omega) = \arg \min_{q \in \mathcal{M}} \frac{1}{2} \sum_{j=1}^k \varphi_j(\omega) \text{dist}(q, p_j)^2 = \arg \min_{q \in \mathcal{M}} L(q, \omega)$$

that satisfies the interpolation conditions (1a), (1b). The requirement to meet the sampled derivatives $v_j^i = \partial_i f(\omega_j)$ entails conditions on the partial derivatives of the weight functions $\varphi_j(\omega)$. We work under the general assumption that the interpolation procedure takes place on a domain, where the weighted Riemannian barycenters exist and are unique. For a detailed analysis of interpolation via Riemannian barycenters, we refer to Sander 2012, 2016 and Grohs, Harder, and Sander 2015.

3.1. TRACKING THE BARYCENTERS VIA THE IMPLICIT FUNCTION THEOREM

Introduce the parametric gradient field

$$G: \mathcal{M} \times \mathbb{R}^d \rightarrow T\mathcal{M}, \quad (q, \omega) \mapsto G(q, \omega) := \text{grad}_q L(q, \omega) = - \sum_{j=1}^k \varphi_j(\omega) \text{Log}_q(p_j) \quad (6)$$

and note that G is a smooth vector field on the product manifold $\mathcal{M} \times \mathbb{R}^d$. Suppose that G vanishes at (q^*, ω^*) . Our strategy is to parameterize the zero-sets locally via the implicit function theorem. By differentiating the corresponding implicit function, we will establish a relation between the derivatives of the interpolation weight functions $\omega \mapsto \varphi_i(\omega)$ of (2) and the sampled derivatives (1b).

The implicit function theorem has a close relative, the inverse function theorem, and both theorems rely on a full-rank condition for a certain differential. While a manifold version of the inverse function theorem appears in many standard textbooks on differential geometry (e.g., Lee 2012, Thm 4.5), we were not able to locate a textbook reference for a manifold counterpart to the implicit function theorem. Yet, because both \mathcal{M} and $T\mathcal{M}$ are differentiable manifolds, it is straightforward to transfer the classical implicit function theorem, and often, it is simply taken for granted (as in Sander 2012,

Thm. 2.2, Sander 2016, Thm 4.1.). The interesting part is how the rank condition transforms. This is the contents of the following lemma.

Lemma 3.1. *Let \mathcal{M} be a Riemannian manifold.*

1. *Let $G: \mathcal{M} \rightarrow T\mathcal{M}$ be a smooth vector field. At locations, where G vanishes, the differential of G coincides with the covariant derivative, i.e., at $q^* \in \mathcal{M}$ with $G(q^*) = 0$, it holds*

$$dG(q^*)[v] = (\nabla_v G)(q^*), \quad \text{for all } v \in T_{q^*}\mathcal{M}.$$

2. *In the special case, where G is the gradient vector field $\text{grad } g: \mathcal{M} \rightarrow T\mathcal{M}$ of a scalar function $g: \mathcal{M} \rightarrow \mathbb{R}$, the Hesse form of g coincides with the differential of $\text{grad } g$ at locations, where the gradient vanishes, i.e., at $q^* \in \mathcal{M}$ with $\text{grad } g(q^*) = 0$, it holds*

$$d(\text{grad } g)(q^*)[v] = \text{Hess } g(q^*)[v], \quad \text{for all } v \in T_{q^*}\mathcal{M}.$$

In particular, the differential $d(\text{grad } g)(q^)$ has full rank, if the Hesse form $\text{Hess } g(q^*)$ has full rank.*

Proof. Let $v \in T_{q^*}\mathcal{M}$. We compute $dG(q^*)[v]$. Let $\gamma: I \rightarrow \mathcal{M}$ be a smooth curve with $\gamma(0) = q^*$, $\dot{\gamma}(0) = v$. On a suitably small neighborhood \tilde{M} around q^* , construct a local, orthonormal frame of vector fields $\{E_i \in X(\mathcal{M}) \mid i = 1, \dots, m\}$ as outlined in Lee 1997, Exercise 3.2, p. 24. This means that we obtain vector fields $E_i: \tilde{M} \rightarrow T\mathcal{M}$, $i = 1, \dots, m$ such that at each $p \in \tilde{M}$, $\{E_i(p) \mid i = 1, \dots, m\}$ is an orthonormal basis of $T_p\mathcal{M}$, i.e.,

$$\langle E_i(p), E_j(p) \rangle_p = \delta_{ij} \quad \text{for all } p \in \tilde{M}.$$

W.l.o.g., assume that the image of γ is contained in \tilde{M} . We express the vector field G along γ in terms of the orthonormal frame

$$(G \circ \gamma)(t) = \sum_i \langle (G \circ \gamma)(t), E_i(\gamma(t)) \rangle_{\gamma(t)} E_i(\gamma(t)),$$

and differentiate according to the product rule (do Carmo 1992, Chapter 2, Prop. 3.2)

$$\begin{aligned} dG(q^*)[v] &= \left. \frac{d}{dt} \right|_{t=0} (G \circ \gamma)(t) \\ &= \sum_i \left[\left. \frac{d}{dt} \right|_{t=0} (\langle (G \circ \gamma)(t), E_i(\gamma(t)) \rangle_{\gamma(t)}) E_i(q^*) \right. \\ &\quad \left. + \langle G(q^*), E_i(q^*) \rangle_{q^*} \left. \frac{d}{dt} \right|_{t=0} (E_i \circ \gamma)(t) \right] \\ &= \sum_i \left(\left\langle \frac{D(G \circ \gamma)}{dt}(0), E_i(q^*) \right\rangle_{q^*} + \langle G(q^*), \frac{D(E_i \circ \gamma)}{dt}(0) \rangle_{q^*} \right) E_i(q^*) \\ &= \sum_i \langle (\nabla_v G)(q^*), E_i(q^*) \rangle_{q^*} E_i(q^*) = (\nabla_v G)(q^*). \end{aligned}$$

Note that we used twice that $G(q^*) = 0$. If G is the gradient vector field of a scalar function $G = \text{grad } g$, then the latter becomes the Hessian, $(\nabla_v \text{grad } g)(q^*) = \text{Hess } g(q^*)[v]$. \square

By Lemma 3.1, the condition on the rank of the differential of a vector field translates to a condition on the covariant derivative (at points, where the vector field vanishes). This makes it straightforward to transfer any of the classical proofs of the implicit function theorem to the manifold setting.

Theorem 3.2 (An implicit function theorem for vector fields). *Let \mathcal{M} be an m -dimensional Riemannian manifold and let $G: \mathcal{M} \times \mathbb{R}^d \rightarrow T\mathcal{M}, (q, \omega) \mapsto G(q, \omega) \in T_q\mathcal{M}$ be a smooth mapping such that both $q \mapsto G(q, \omega) \in T_q\mathcal{M}$ with ω considered as fixed and $\omega \mapsto G(q, \omega) \in T_q\mathcal{M}$ with q considered as fixed are smooth vector fields. Let $(q^*, \omega^*) \in \mathcal{M} \times \mathbb{R}^d$ be a point such that $G(q^*, \omega^*) = 0 \in T_{q^*}\mathcal{M}$ and let $q \mapsto G^*(q) := G(q, \omega^*)$. Assume that the linear form $(\nabla_{(\cdot)}G^*)(q^*): T_{q^*}\mathcal{M} \rightarrow T_{q^*}\mathcal{M}, v \mapsto (\nabla_v G^*)(q^*)$ has full rank m .*

Then there exists an open domain $\Omega \subset \mathbb{R}^d$ around ω^ and a differentiable function $\hat{f}: \Omega \rightarrow \mathcal{M}, \omega \mapsto \hat{f}(\omega)$ such that*

$$\Omega \ni \omega \mapsto G(\hat{f}(\omega), \omega) \equiv 0 \in T_{\hat{f}(\omega)}\mathcal{M}. \quad (7)$$

Proof. Use the standard trick to obtain the implicit function theorem from the inverse function theorem see, e.g., Lee 2012, Thm. C.4o. To this end, consider the auxiliary function

$$\gamma: \mathcal{M} \times \mathbb{R}^d \rightarrow T\mathcal{M} \times \mathbb{R}^d. \quad (q, \omega) \mapsto \begin{pmatrix} \Gamma_1(q, \omega) \\ \Gamma_2(q, \omega) \end{pmatrix} = \begin{pmatrix} G(q, \omega) \\ \omega \end{pmatrix}.$$

Note that Γ is a smooth map between differentiable product manifolds. By Lemma 3.1, the differential of the first component function of Γ by q at the point (q^*, ω^*) is given by $d_q G(q^*, \omega^*)[v] = (\nabla_v G^*)(q^*)$ and thus has full rank by assumption. The differential of the second component function by q vanishes, while its differential by ω is the identity on \mathbb{R}^d . As a consequence, the differential of Γ at (q^*, ω^*) has full rank and the inverse function theorem yields the existence of a local inverse $\Gamma^{-1} = H = (H_1, H_2)$. It holds

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \Gamma(H(v, \omega)) = \begin{pmatrix} G((H_1(v, \omega), H_2(v, \omega))) \\ H_2(v, \omega) \end{pmatrix}.$$

In particular, $H_2(v, \omega) = \omega$ and the function $\hat{f}: \omega \mapsto H_1(0, \omega)$ is the sought-after implicit function with $G(\hat{f}(\omega), \omega) = G(H_1(0, \omega), H_2(v, \omega)) \equiv 0$ on a local domain, where \hat{f} is defined. \square

An alternative proof that works with local coordinates is given in Séguin and Kressner 2022, Theorem 3.1. However, this proof omits the technical detail stated in Lemma 3.1.

In the setting of Theorem 3.2, implicit differentiation of (7) along a curve $c: I \rightarrow \Omega$ with $c(0) = \omega, \dot{c}(0) = w$ yields

$$\begin{aligned} 0 &= \left. \frac{d}{dt} \right|_{t=0} G(\hat{f}(c(t)), c(t)) = \left(d_q G((\hat{f}(\omega), \omega)), d_\omega G((\hat{f}(\omega), \omega)) \right) \begin{bmatrix} d\hat{f}_\omega[w] \\ w \end{bmatrix} \\ &= d_q G(\hat{f}(\omega), \omega) \left[d\hat{f}(\omega)[w] \right] + d_\omega G(\hat{f}(\omega), \omega)[w] \end{aligned}$$

Since $G(\hat{f}(\omega), \omega) = 0$, Lemma 3.1 applies and yields

$$\left(\nabla_{d\hat{f}(\omega)[w]} G(\cdot, \omega) \right) (\hat{f}(\omega)) = -d_\omega G(\hat{f}(\omega), \omega)[w] \in T_{\hat{f}(\omega)}\mathcal{M}. \quad (8)$$

Similar considerations were made in Sander 2016, Section 6, but for a different purpose.

3.2. A RELATION BETWEEN THE DERIVATIVES OF THE INTERPOLATION WEIGHT FUNCTIONS AND THE SAMPLED DERIVATIVES

Now, we apply the results of Subsection 3.1 to the specific gradient field G of (6). Recall that G is the gradient vector field (with respect to q) of the scalar function $L(q, \omega) = \frac{1}{2} \sum_{j=1}^k \varphi_j(\omega) \text{dist}(q, p_j)^2$ so that $G(q, \omega) := \text{grad}_q L(q, \omega) = -\sum_{j=1}^k \varphi_j(\omega) \text{Log}_q(p_j)$. By construction, G vanishes at the sample locations $q = p_l, \omega = \omega_l, j = 1, \dots, k$. Moreover, by Theorem 2.4 and (5), the Hessians at the sample locations have full rank. Theorem 3.2 yields locally an implicit parameterization of the zero set of the gradient field around each sample point. Since the sought-after interpolant is determined via an optimization problem, and thus, in turn, via the zero set of the gradient field, the interpolant coincides with the implicit function \hat{f} . In particular, $\hat{f}(\omega_l) = p_l$.

Fix a sample point $(\omega_l, p_l), l \in 1, \dots, k$. Let $\{e_i \in \mathbb{R}^d \mid i = 1, \dots, d\}$ denote the Cartesian unit vectors in \mathbb{R}^d . With $w = e_i, \omega = \omega_l$, eq. (8) relates the partial derivatives $v_l^i := \partial_i \hat{f}(\omega_l) = \text{d}\hat{f}(\omega_l)[e_i]$ of \hat{f} to the partial derivatives of the interpolation weight functions,

$$\begin{aligned} \left(\nabla_{v_l^i} \text{grad}_q L(p_l, \omega_l) \right) (p_l) &= -\text{d}_\omega G(p_l, \omega_l)[e_i] \Leftrightarrow \\ \text{Hess}_q L(p_l, \omega_l)[v_l^i] &= v_l^i = \sum_{j=1}^k \partial_i \varphi_j(\omega_l) \text{Log}_{p_l}(p_j) = \sum_{j=1, j \neq l}^k \partial_i \varphi_j(\omega_l) \text{Log}_{p_l}(p_j). \end{aligned}$$

This is in direct correspondence with (22) in the Euclidean setting, see Appendix A in the supplements.

In order to be able to match any prescribed, sampled partial derivatives $v_l^i = \partial_i \hat{f}(\omega_l)$, we require the sets

$$\mathcal{L}_l := \{\text{Log}_{p_l}(p_j) \mid j = 1, \dots, k, j \neq l\} \subset T_{p_l} \mathcal{M}, \quad l = 1, \dots, k$$

to span the full tangent space $T_{p_l} \mathcal{M}$. Then any choice of interpolation weight function φ_j with $\varphi_j(\omega_l) = \delta_{jl}$ that satisfies

$$v_l^i = \sum_{j=1, j \neq l}^k \partial_i \varphi_j(\omega_l) \text{Log}_{p_l}(p_j) \quad (9)$$

yields an interpolant \hat{f} that satisfies the interpolation conditions (1a), (1b).

Remark 3.3. Recall that $\dim(\mathcal{M}) = m$. For every fixed index $l \in \{1, \dots, k\}$, let $\{E_j^l \mid j = 1, \dots, m\} \subset T_{p_l} \mathcal{M}$ be a local orthonormal frame. Then, each of the tangent vectors $\text{Log}_{p_l}(p_j) \in T_{p_l} \mathcal{M}$ features a representation with m coordinate coefficients

$$\text{Log}_{p_l}(p_j) = x_l^j(1)E_1^l + \dots + x_l^j(m)E_m^l.$$

Likewise for the sampled derivatives,

$$v_l^i = \alpha_l^i(1)E_1^l + \dots + \alpha_l^i(m)E_m^l.$$

With respect to the coordinates of the local frame, the unknowns in (9), i.e., the scalar coefficients $\partial_i \varphi_j(\omega_l)$,

are determined by

$$\begin{pmatrix} x_l^1(1) & \dots & x_l^{l-1}(1) & x_l^{l+1}(1) & \dots & x_l^k(1) \\ \vdots & & \vdots & \vdots & & \vdots \\ x_l^1(m) & \dots & x_l^{l-1}(m) & x_l^{l+1}(m) & \dots & x_l^k(m) \end{pmatrix} \begin{pmatrix} \partial_i \varphi_1(\omega_l) \\ \vdots \\ \partial_i \varphi_{l-1}(\omega_l) \\ \partial_i \varphi_{l+1}(\omega_l) \\ \vdots \\ \partial_i \varphi_k(\omega_l) \end{pmatrix} = \begin{pmatrix} \alpha_l^i(1) \\ \vdots \\ \vdots \\ \alpha_l^i(m) \end{pmatrix}. \quad (10)$$

The system matrix in (10) is of dimensions $m \times (k-1)$. In order to ensure that solutions exist, the number of sample points must be larger than the manifold dimension, $k > m$, and the logs $\text{Log}_{p_l}(p_j)$, $j \neq k$ must span the tangent space $T_{p_l}\mathcal{M}$. If $k > m+1$, we have more coefficients $\partial_i \varphi_j(\omega_l)$ than are needed for a representation of v_l^i in terms of the logs. The system is underdetermined. The additional degrees of freedom may be used to impose extra conditions, e.g. that the derivative coefficients sum up to zero. In fact, this is required for consistency, because $\sum_j \varphi_j(\omega) \equiv 1$, i.e., the weights correspond to a discrete signed measure of unit weight.

4. MULTIVARIATE HERMITE INTERPOLATION IN THE TANGENT SPACE

As an alternative to BHI, multivariate Hermite interpolation on a manifold may also be conducted in a straightforward manner by moving all data, i.e., sample points and derivatives, to a selected tangent space and to perform the interpolation therein. The following steps detail this approach.

1. Choose a manifold location $q_0 \in \mathcal{M}$ that is to act as the ‘data center’. For example, q_0 may be one of the sample points or it may be the Riemannian barycenter of the given sample data set.
2. Map all sampled manifold locations to the tangent space at q_0 , i.e., compute $w_j = \text{Log}_{q_0}(p_j) \in T_{q_0}\mathcal{M}$, $j = 1, \dots, k$.
3. Transport the sampled velocity vectors to the selected tangent space $T_{q_0}\mathcal{M}$. To this end, compute for $i = 1, \dots, d$, $l = 1, \dots, k$,

$$\begin{aligned} T_{q_0}\mathcal{M} \ni \hat{v}_l^i &= d(\text{Log}_{q_0})_{p_l}[\partial_i f(\omega_l)] \\ &\approx \frac{\text{Log}_{q_0}(\text{Exp}_{p_l}(\Delta t \partial_i f(\omega_l))) - \text{Log}_{q_0}(\text{Exp}_{p_l}(-\Delta t \partial_i f(\omega_l)))}{2\Delta t}. \end{aligned} \quad (11)$$

The latter finite-differences approximation can be used in cases, where the Riemannian log map is not available in closed form, for details and remarks on the numerical accuracy, see Zimmermann 2020.

4. With all data gathered on one and the same tangent space, classical Hermite interpolation in vector spaces can be pursued. The interpolant is constructed as a weighted linear combination of the sampled data as

$$\hat{f}_{\text{tan}}(\omega) = \sum_{j=1}^k \varphi_j(\omega) w_j + \sum_{i=1}^d \sum_{l=1}^k \psi_{i,l}(\omega) \hat{v}_l^i \in T_{q_0}\mathcal{M}. \quad (12)$$

5. The sampled data provides Hermite sample values for the weight functions $\varphi_j(\omega), \psi_{i,l}(\omega)$. Let $\vec{\varphi}_j := (\varphi_j(\omega_1), \dots, \varphi_j(\omega_k))$ be the vector of sampled values of the j th coefficient function $\omega \mapsto \varphi_j(\omega)$. Likewise, let $\vec{\psi}_{i,l} := (\psi_{i,l}(\omega_1), \dots, \psi_{i,l}(\omega_k))$. For each of the coefficient functions, one needs to solve a multiple-input scalar-output Hermite interpolation problem, where the sample locations are $\omega_1, \dots, \omega_k$ and the sample values for the various coefficient functions are

$$\begin{pmatrix} \vec{\varphi}_j \\ \partial_1 \vec{\varphi}_j \\ \vdots \\ \partial_d \vec{\varphi}_j \end{pmatrix} = \begin{pmatrix} e_j \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \quad \begin{pmatrix} \vec{\psi}_{1,l} \\ \partial_1 \vec{\psi}_{1,l} \\ \vdots \\ \partial_d \vec{\psi}_{1,l} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ e_l \\ \vdots \\ \mathbf{0} \end{pmatrix}, \dots, \begin{pmatrix} \vec{\psi}_{d,l} \\ \partial_1 \vec{\psi}_{d,l} \\ \vdots \\ \partial_d \vec{\psi}_{d,l} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ e_l \end{pmatrix} \in \mathbb{R}^{k(d+1)}. \quad (13)$$

Here, the data is written in vector packs of size k (k values from $\varphi_j(\omega)$ evaluated at the k sample locations, k values from $\partial_1 \varphi_j(\omega)$ evaluated at the k sample locations, etc.) and $e_j = (0, \dots, \overset{j}{1}, \dots, 0)^T \in \mathbb{R}^k$ denotes the j th canonical unit vector.

6. The interpolants of the coefficient functions can now be obtained from any multivariate interpolation scheme that is able to tackle Hermite data, e.g., the gradient-enhanced Kriging method, Forrester, Sobester, and Keane 2008, Section 7.2. In summary, this requires to solve $k(d+1)$ Euclidean interpolation problems. The resulting tangent vector of (12) is mapped back to the manifold via

$$\hat{f}(\omega) = \text{Exp}_{q_0}(\hat{f}_{\text{tan}}(\omega)) \in \mathcal{M}. \quad (14)$$

Preliminary comparison. For simplicity, the above interpolation approach is referred to as the tangent space Hermite interpolation (**THI**) method. The BHI method outlined in Section 3 works with a number of k weight coefficient functions with properly adjusted partial derivatives, which are obtained by solving the linear equation systems (10). The THI method works with $k(d+1)$ coefficient functions, each of which comes with its own Hermite sample data set. At the preprocessing stage of the BHI approach, the Riemannian logarithm function needs to be queried. In contrast, prior to conducting THI, one needs to query the Riemannian logarithm as well as its total derivative (or an approximation thereof). In the online stage of BHI, the values of the coefficient functions at an untried location $\omega^* \in \mathbb{R}^d$ are determined via a Riemannian optimization problem. In fact, it is the exact same Riemannian optimization problem that needs to be tackled for performing non-Hermite barycentric interpolation, except that the weight functions are properly adjusted a priori. The practitioner may choose any preferred method from the zoo of optimization algorithms, see, e.g., Absil, Mahony, and Sepulchre 2008, and retractions could be used to replace the Riemannian exponential. For THI, the values of the coefficient functions at ω^* are obtained via solving $k(d+1)$ Euclidean interpolation problems in a selected tangent space. The back-mapping to the manifold must be via the Riemannian exponential, for a retraction would compromise the interpolation condition, unless the inverse of the same retraction is also used at the preprocessing stage.

Table 1 summarizes the main features of both approaches.

	Barycentric	Tangent space
#(coefficient functions)	k	$k(d+1)$
minimum #(samples)	$k > \dim(\mathcal{M}) + 1$	no restriction
base-point dependent	no	yes
manifold data processing via	Log_q	$\text{Log}_q, d(\text{Log}_q)_p$
model evaluation via	Riemann. optimization	interp. in tangent space

Table 1: Comparison of the barycentric and the tangent space approach to Hermite manifold interpolation for a d -variate interpolation problem with a number of k sample locations $\omega_j \in \mathbb{R}^d$.

5. NUMERICAL EXAMPLES

In this section, we demonstrate multivariate Hermite manifold interpolation by means of academic examples. For clarity of the exposition, but for no mathematical reasons, we focus in the experiments on the case, where \mathcal{M} is an embedded submanifold of \mathbb{R}^N and we express data points $p \in \mathcal{M}$ and tangent vectors $v \in T_p\mathcal{M}$ in *extrinsic coordinates*. This means that we use coordinates of \mathbb{R}^N rather than local coordinates to address locations on \mathcal{M} . This situation is also to be expected in practical applications.

Example. On the unit sphere $\mathcal{M} = S^2 = \{p \in \mathbb{R}^3 \mid p_1^2 + p_2^2 + p_3^2 = 1\} \subset \mathbb{R}^3$, which is of (manifold) dimension $m = 2$, points are addressed by their three Euclidean coordinates $p = (p_1, p_2, p_3)^T$. Likewise, tangent vectors $v \in T_p S^2 = \langle p \rangle^\perp$ are also given by extrinsic coordinates $v = (v_1, v_2, v_3)^T \in T_p S^2 \subset \mathbb{R}^3$.

Consider a differentiable, d -variate, \mathcal{M} -valued function

$$f: \mathbb{R}^d \supset D \rightarrow \mathcal{M} \subset \mathbb{R}^N, \omega \mapsto f(\omega)$$

Suppose that k sample locations $\omega^j = (\omega_1^j, \dots, \omega_d^j)^T \in \mathbb{R}^d$ are selected and that sample points $p^j = (p_1^j, \dots, p_N^j)^T = f(\omega^j) \in \mathcal{M} \subset \mathbb{R}^N$ and tangent vectors $v_i^j = \partial_i f(\omega^j) \in T_{p^j} \mathcal{M} \subset \mathbb{R}^N, i = 1, \dots, d$ are available.

For THI, the coefficient weight functions are obtained by Hermite interpolation of the data sets listed in (13). In the BHI approach, we construct interpolants by computing numerical solutions to the Riemannian optimization problem (2). In this case, the vector of sample values for the j th weight function φ_j in (2) is

$$(\varphi_j(\omega^1), \dots, \varphi_j(\omega^k)) = e_j = (0, \dots, 0, \overset{j}{1}, 0, \dots, 0) \in \mathbb{R}^k \quad (15)$$

and is independent of the sample points $p_j \in \mathcal{M}$. The vector of the partial derivatives of φ_j at the sample locations is obtained from Algorithm 2.

Algorithm 2. Practical computation of the derivatives of the weight functions for BHI.

Input: Hermite data set consisting of sample pairs $(p_j = f(\omega_j), \omega_j) \in \mathcal{M} \times \mathbb{R}^d, j = 1, \dots, k$, and partial derivatives $v_l^i = \partial_l f(\omega_l) \in T_{p_l} \mathcal{M}, i = 1, \dots, d, l = 1, \dots, k$.

```

1: for  $i = 1, \dots, d$  do
2:   for  $l = 1, \dots, k$  do
3:     Compute  $\text{Log}_{p_l}(p_j) \in T_{p_l}\mathcal{M} \subset \mathbb{R}^N$ ,  $j = 1, \dots, k$ ,  $j \neq l$ 
4:      $X_l := \left( \text{Log}_{p_l}(p_1) \mid \dots \mid \text{Log}_{p_l}(p_{l-1}) \mid \text{Log}_{p_l}(p_{l+1}) \mid \dots \mid \text{Log}_{p_l}(p_k) \right) \in \mathbb{R}^{N \times (k-1)}$ 
5:     Solve  $\begin{pmatrix} X_l \\ \mathbf{1}, \dots, \mathbf{1} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_{k-1} \end{pmatrix} = \begin{pmatrix} v_l^i \\ 0 \end{pmatrix}$ 
6:      $(\partial_i \varphi_1(\omega_l), \dots, \partial_i \varphi_{l-1}(\omega_l), \partial_i \varphi_l(\omega_l), \partial_i \varphi_{l+1}(\omega_l), \dots, \partial_i \varphi_k(\omega_l)) := (c_1, \dots, c_{l-1}, 0, c_l, \dots, c_{k-1})$ 
7:   end for
8: end for

```

Output: Partial derivatives of the weight interpolation functions at the sample locations $\partial_i \varphi_j(\omega_l)$, $j, l = 1, \dots, k$, $i = 1, \dots, d$.

Remark 5.1. In the upcoming experiments, we face the setting, that $\dim(\mathcal{M}) = m < k - 1$, where k is the number of sample points. Hence, the linear system in step 5 of Alg. 2 is underdetermined. The bottom row of all ones is added to the system matrix to enforce that $\sum_j \partial_i \varphi_j(\omega_l) = 0$.

The minimum 2-norm solution to the system is obtained via the pseudo-inverse based on the SVD, see Golub and Van Loan 1996, §5.7, p. 270–273. Let $r = \text{rank } X$ and let $U_r \Sigma_r V_r^T = X$ be the reduced SVD, i.e., $U_r \in \mathbb{R}^{N \times r}$, $\Sigma_r \in \mathbb{R}^{r \times r}$, $V_r \in \mathbb{R}^{(k-1) \times r}$. The columns of U_r form a basis for the column space of X . The underdetermined equation $Xc = v_l^i$ corresponds to an equation for the coordinates with respect to the basis given by U_r , namely, $(U_r^T X)c = (\Sigma_r V_r^T)c = \tilde{v}_l^i = U_r^T v_l^i$. Writing $\mathbf{1} \in \mathbb{R}^{k-1}$ for the vector with all entries equal to 1, we enforce that the derivatives of the coefficient functions sum up to zero via

$$\tilde{X}_r c := \begin{pmatrix} \Sigma_r V_r^T \\ \mathbf{1}^T \end{pmatrix} c = \begin{pmatrix} \tilde{v}_l^i \\ 0 \end{pmatrix}.$$

The minimum 2-norm solution is obtained via $c = \tilde{X}_r^T (\tilde{X}_r \tilde{X}_r^T)^{-1} \tilde{v}_l^i$. Using the SVD data of X and Schur complement inversion, this can be calculated in closed form

$$c = x + \frac{\langle \mathbf{1}, x \rangle}{s} \left(V_r (V_r^T \mathbf{1}) - \mathbf{1} \right), \quad x = V_r \Sigma_r^{-1} \tilde{v}_l^i \in \mathbb{R}^{k-1}, \quad s = (k-1) - \|V_r^T \mathbf{1}\|_2^2 \in \mathbb{R}. \quad (16)$$

For constructing the interpolation weight functions φ_j in BHI and $\varphi_j, \psi_{i,l}$ in THI, we use the method of gradient-enhanced Kriging with the cubic correlation model and a fixed correlation hyperparameter vector $\theta = (\theta_1, \theta_2) = (0.5, 0.5)$. The cubic correlation model is $\rho(\theta, \omega^i, \omega^j) = \prod_{l=1}^2 \rho_l(\theta_l, (\omega_l^i - \omega_l^j))$, where

$$\rho_l(\theta_l, (\omega_l^i - \omega_l^j)) = \begin{cases} 1 - 3 \left(\theta_l |\omega_l^i - \omega_l^j| \right)^2 + 2 \left(\theta_l |\omega_l^i - \omega_l^j| \right)^3, & \theta_l |\omega_l^i - \omega_l^j| < 1 \\ 0, & \theta_l |\omega_l^i - \omega_l^j| \geq 1 \end{cases}.$$

For the details, we refer to Forrester, Sobester, and Keane 2008, Section 7.2 and Zimmermann 2013. The input sample values are given by (15) and Algorithm 2 for BHI and by (13) for THI, respectively. BHI requires to solve the optimization problem (2) and thus to provide the optimizer with an initial guess. In the first run of the optimizer, we use the first sample location as the initial guess, i.e., $q_0 = p_1$. For every consecutive run, we use the optimized solution q^* from the previous run as the starting point for the next optimization procedure.

5.1. INTERPOLATION OF THE GAUSS MAP OF THE HELICOID

In this section we consider an academic example on the unit sphere $\mathcal{M} = S^2$. The Riemannian exponential and logarithmic maps on S^2 are

$$\begin{aligned}\text{Exp}_q(v) &= \cos(\|v\|_2)q + \sin(\|v\|_2) \frac{v}{\|v\|_2} \in S^2, \\ \text{Log}_q(p) &= \arccos(\langle q, p \rangle) \frac{p - \langle q, p \rangle q}{\|p - \langle q, p \rangle q\|_2} \in T_q S^2,\end{aligned}$$

respectively. As a test function, we use the Gauß map of the helicoid in \mathbb{R}^3 ,

$$f: [a, b]^2 \rightarrow S^2, \quad (\omega_1, \omega_2) \mapsto \frac{1}{(e^{2\omega_1} + 1)} \begin{pmatrix} 2e^{\omega_1} \cos(\omega_2) \\ 2e^{\omega_1} \sin(\omega_2) \\ e^{2\omega_1} - 1 \end{pmatrix}.$$

This function yields the normal field of the helicoid and is obtained from the stereographic projection of $(\omega_1, \omega_2) \mapsto \exp(\omega_1 + i\omega_2) \in \mathbb{C}$ onto S^2 .

The partial derivatives of f are

$$\begin{aligned}\partial_1 f(\omega_1, \omega_2) &= \frac{-2e^{2\omega_1}}{(e^{2\omega_1} + 1)^2} \begin{pmatrix} 2e^{\omega_1} \cos(\omega_2) \\ 2e^{\omega_1} \sin(\omega_2) \\ e^{2\omega_1} - 1 \end{pmatrix} + \frac{2}{e^{2\omega_1} + 1} \begin{pmatrix} e^{\omega_1} \cos(\omega_2) \\ e^{\omega_1} \sin(\omega_2) \\ e^{2\omega_1} \end{pmatrix}, \\ \partial_2 f(\omega_1, \omega_2) &= \frac{1}{e^{2\omega_1} + 1} \begin{pmatrix} -2e^{\omega_1} \sin(\omega_2) \\ 2e^{\omega_1} \cos(\omega_2) \\ 0 \end{pmatrix}.\end{aligned}$$

We sample the function values and partial derivatives of f on $k = 9$ uniformly distributed sample locations $(\omega_1^j, \omega_2^j) \in [a, b]^2 = [-\frac{1}{4}\pi, \frac{1}{4}\pi]^2$, $j = 1, \dots, 9$ and conduct BHI and THI.

The numerical gradient descent method of Algorithm 1 is conducted with a fixed step size of $\alpha = 1.0$ and a numerical convergence threshold of $\tau = 1.0 \cdot 10^{-8}$. For THI, the Riemannian barycenter of the sample data set is used as the tangent space base point. The respective interpolant is evaluated on a uniform grid of 101×101 points in the same domain $[a, b]^2$.

Table 2 summarizes the parameter settings for this experiment and lists the wall clock time and the interpolation errors. The wall clock time is split into an offline stage and an online stage. The offline stage consists of the construction of the Hermite interpolants for the weight coefficient functions φ_j for BHI and $\varphi_j, \psi_{i,l}$ for THI, respectively. The online stage accounts for querying the interpolant at a given parameter location. The evaluation times are averaged over the number of 10, 201 = 101^2 runs. The sample data set in form of surface normal vectors, their partial derivatives and the interpolation error surfaces are displayed in Figure 1.

As can be seen from the Table 2 and Figure 1, the averaged accuracy of the THI interpolant is roughly a factor of 6 times better than the averaged accuracy of the BHI interpolant. A finite difference check shows that both methods provide interpolants that meet the sampled tangent vectors. The computation time for the offline stage under the THI approach is 3.5 times higher than that for BHI. This is

Parameter settings					
Manifold	#variables	domain	#samples	threshold	line step
S^2	$d = 2$	$[-\frac{1}{4}\pi, \frac{1}{4}\pi]^2$	$k = 9$	$\tau = 1.0 \cdot 10^{-8}$	$\alpha = 1.0$

Results: barycentric Hermite interpolation (BHI)					
Wall clock time		Interpolation error		FD error	
offline	online	max	avg	avg. $\partial_1 \hat{f}(\omega^j)$	avg. $\partial_2 \hat{f}(\omega^j)$
$7.2 \cdot 10^{-3} \text{s}$	$6.42 \cdot 10^{-5} \text{s}$	$2.14 \cdot 10^{-2}$	$8.23 \cdot 10^{-3}$	$6.17 \cdot 10^{-6}$	$6.27 \cdot 10^{-6}$

Results: tangent space Hermite interpolation (THI)					
Wall clock time		Interpolation error		FD error	
offline	online	max	avg	avg. $\partial_1 \hat{f}(\omega^j)$	avg. $\partial_2 \hat{f}(\omega^j)$
$2.5 \cdot 10^{-2} \text{s}$	$1.5 \cdot 10^{-4} \text{s}$	$2.36 \cdot 10^{-3}$	$1.10 \cdot 10^{-3}$	$6.47 \cdot 10^{-6}$	$7.61 \cdot 10^{-6}$

Table 2: Parameter settings and interpolation results for the experiment of Subsection 5.1 associated with Figure 1. The ‘offline’ time refers to the construction of the interpolant from the sampled data, while the ‘online’ time accounts for querying the interpolant at a trial location.

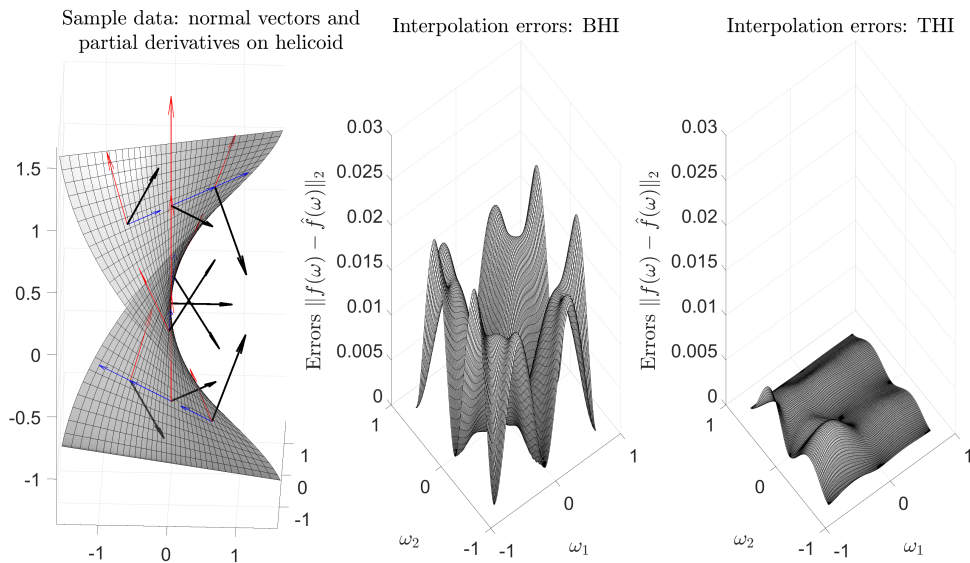


Figure 1: Interpolating the Gauß map of the Helicoid from Subsection 5.1. Left: sample data set and partial derivatives. Middle: error surface obtained with BHI. Right: error surface obtained with THI.

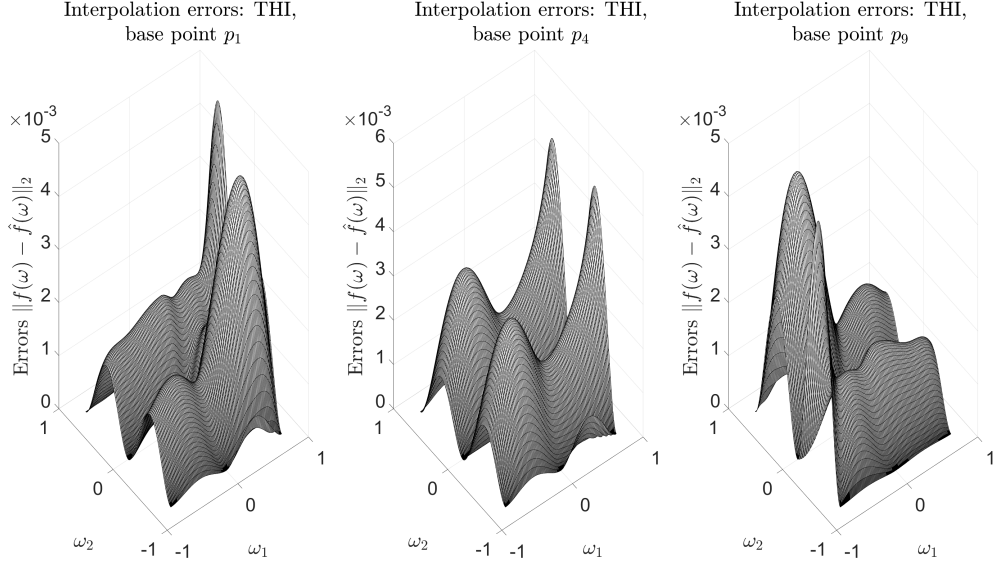


Figure 2: Error surfaces under the THI method from Subsection 5.1 for different base tangent spaces.

expected, since for THI, the data preprocessing involves evaluating the Riemann exp and log maps plus the differential of the latter for moving the sampled derivatives to the same tangent space.³ Moreover, a number of $k(d+1) = 27$ Euclidean Hermite interpolation problems have to be solved to obtain the interpolated weight functions. In contrast, for BHI, only $k = 9$ coefficient functions have to be fitted. This, however, involves solving an underdetermined linear equation system.

In the online stage of BHI, the basic gradient descent converges after an average number of 4.9 iterations with the worst case taking 7 iterations. On average, for the test case at hand, BHI is online 2.3 times faster than THI. However, too much weight should not be given to the timing results, as they are highly dependent on the chosen optimization algorithm and choosing a ‘best one’ is beyond the scope of these experiments.

The THI method produces different interpolants for different choices of the tangent space base point. To illustrate this issue, Figure 2 shows the error surfaces corresponding to three different base-point selections. The left picture Figure 2 is produced by working in $T_{p_1}S^2$, the middle picture by working in $T_{p_4}S^2$, and the right picture corresponds to working in $T_{p_9}S^2$, where $p_j = f(\omega_j)$.

5.2. INTERPOLATION ON $SO(3)$

The next test case considers a Hermite data set on the special orthogonal group $SO(3) = \{Q \in \mathbb{R}^{3 \times 3} \mid Q^T Q = I, \det(Q) = 1\}$. The tangent space of $SO(3)$ at a point $Q \in SO(3)$ is $T_Q SO(3) = \{V \in \mathbb{R}^{3 \times 3} \mid Q^T V + V^T Q = 0\}$. The Riemannian exp and log maps on $SO(3)$ are

$$\begin{aligned} \text{Exp}_Q: T_Q SO(3) &\rightarrow SO(3), & V &\mapsto Q \exp_m(Q^T V), \\ \text{Log}_Q: \mathcal{D}_Q &\rightarrow T_Q SO(3), & R &\mapsto Q \log_m(Q^T R), \end{aligned}$$

³For the academic case at hand, a closed form calculation of $d(\text{Log}_q)_p$ is possible. However, for consistency with the general case, in the numerical experiments, we work with the finite difference approximation of (11).

<i>Parameter settings</i>					
Manifold	#variables	domain	#samples	threshold	line step
$SO(3)$	$d = 2$	$[0.5, 0.5]^2$	$k = 49$ (Cheby.)	$\tau = 1.0 \cdot 10^{-6}$	$\alpha = 1.0$

<i>Results: barycentric Hermite interpolation (BHI)</i>					
Wall clock time		Interpolation error		FD error	
offline	online	max	avg	avg. $\partial_1 \hat{f}(\omega^j)$	avg. $\partial_2 \hat{f}(\omega^j)$
0.41s	0.077s	0.029	0.0069	$4.45 \cdot 10^{-4}$	$4.91 \cdot 10^{-4}$

<i>Results: tangent space Hermite interpolation (THI)</i>					
Wall clock time		Interpolation error		FD error	
offline	online	max	avg	avg. $\partial_1 \hat{f}(\omega^j)$	avg. $\partial_2 \hat{f}(\omega^j)$
0.73s	0.0023s	0.027	0.0065	$3.9 \cdot 10^{-4}$	$4.6 \cdot 10^{-4}$

Table 3: Parameter settings and interpolation results for the experiment of Subsection 5.2 associated with Figure 4. The ‘offline’ time refers to the construction of the interpolant from the sampled data, while the ‘online’ time accounts for querying the interpolant at a trial location.

where \exp_m and \log_m denote the classical matrix exponential and logarithm functions, see Higham 2008, and $\mathcal{D}_Q \subset SO(3)$ is a domain around Q such that for all $R \in \mathcal{D}_Q$, the orthogonal matrix $Q^T R$ does not feature $\lambda = -1$ as an eigenvalue. As a test function, we consider

$$f: [a, b]^2 \rightarrow SO(3), \quad (\omega_1, \omega_2) \mapsto \exp_m X(\omega_1, \omega_2), \text{ where}$$

$$X(\omega_1, \omega_2) = \begin{pmatrix} 0 & \omega_1^2 + \frac{1}{2}\omega_2 & \sin(4\pi(\omega_1^2 + \omega_2^2)) \\ -\omega_1^2 - \frac{1}{2}\omega_2 & 0 & \omega_1 + \omega_2^2 \\ -\sin(4\pi(\omega_1^2 + \omega_2^2)) & -\omega_1 - \omega_2^2 & 0 \end{pmatrix}.$$

The sample locations $P_j = \exp_m X(\omega^j)$ at $\omega^j = (\omega_1^j, \omega_2^j)$ and the corresponding partial derivatives $V_j^i = \frac{d}{dt} \Big|_{t=0} \exp_m(X(\omega^j + t e_i)) = d(\exp_m)(X(\omega^j))[\partial_i X(\omega^j)]$, $i = 1, 2$ of the test function can be obtained by Mathias’ theorem, see Higham 2008, Thm. 3.6:

$$\exp_m \begin{pmatrix} X(\omega^j) & \partial_i X(\omega^j) \\ 0 & X(\omega^j) \end{pmatrix} = \begin{pmatrix} \exp_m(X(\omega^j)) & d(\exp_m)(X(\omega^j))[\partial_i X(\omega^j)] \\ 0 & \exp_m(X(\omega^j)) \end{pmatrix}.$$

For BHI, the partial derivatives of the interpolation weight functions φ_j are computed with Algorithm 2, where we use the vectorized tangent matrices $\text{vec}(\text{Log}_{P_1}(P_j)) \in \mathbb{R}^{3 \cdot 3}$ as columns to form the matrices X_l . The input V_l^i on the right hand side of the equation system in step 5 of Algorithm 2 is vectorized accordingly. For BHI, for each trial location ω^* , the interpolant $\hat{f}(\omega^*)$ is computed with Algorithm 1, while for THI, it is computed according to (12) and (14). For THI, the Riemannian barycenter of the sample data set is used as the tangent space base point.

For quantifying the accuracy of the interpolation, we compute the relative errors $\frac{\|f(\omega) - \hat{f}(\omega)\|_F}{\sqrt{3}}$. Mind that $\|Q\|_F = \sqrt{3}$ for any matrix $Q \in SO(3)$. In this experiment, we rely on a two-dimensional Chebyshev sample plan

$$\left\{ \frac{1}{2}(b-a) \cos\left(\frac{(2j-1)\pi}{2k}\right) + \frac{1}{2}(b+a) \mid j = 1, \dots, k \right\}^2 \subset \mathbb{R}^2.$$

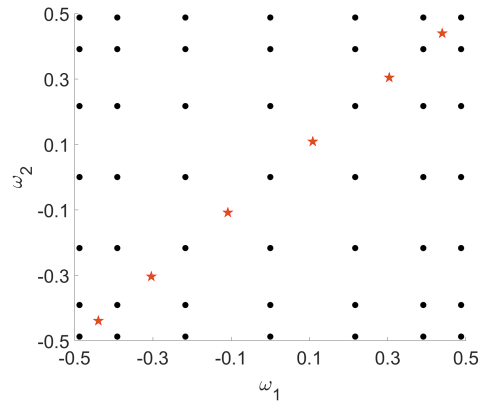


Figure 3: Sample plan associated with the experiment of Section 5.2. Black dots: Chebyshev 7×7 grid on the domain $[-0.5, 0.5]^2$. The stars indicate the trial locations that are used for visualization purposes in the upcoming Figure 7.

We sample the function values and partial derivatives of f on $k = 49$ Chebyshev sample locations $(\omega_1^j, \omega_2^j) \in [a, b]^2 = [-0.5, 0.5]^2$, $j = 1, \dots, 49$, see Figure 3. The numerical gradient descent method of Algorithm 1 is conducted with a fixed step size of $\alpha = 1.0$ and a numerical convergence threshold of $\tau = 1.0 \cdot 10^{-6}$. The BHI and THI interpolants are evaluated on a uniform grid of 76×76 trial points in the same domain. The corresponding interpolation error surfaces are displayed in Figure 4. Table 3 summarizes the parameter settings as well as the timing and accuracy results.

In the test case at hand, BHI and THI produce interpolants of roughly the same averaged accuracy, with THI performing slightly better than BHI. To the naked eye, the error surfaces look identical for both approaches. A finite difference check shows again that both methods provide interpolants that meet the sampled tangent vectors. The computation time for the offline stage under the THI approach is roughly 2 times higher than that for BHI. In the online stage of BHI, the basic gradient descent converges after an average number of 18.0 iterations with the worst case taking 87 iterations. On average, for the test case at hand, THI is online 33 times faster than BHI. We emphasize again that the timing results are expected to vary considerably depending on which optimization algorithm is employed.

In this test case, each interpolated value $\hat{P} = \hat{f}(\omega)$ is a matrix in $SO(3)$. In order to further visualize the interpolation results, Figure 5 displays the interpolated matrix component functions $\omega \mapsto \hat{P}_{11}$, $\omega \mapsto \hat{P}_{22}$, $\omega \mapsto \hat{P}_{33}$, and $\omega \mapsto \hat{P}_{31}$. Because the plots for THI and BHI virtually coincide, only the results for BHI are shown. The first diagonal component of the interpolant is juxtaposed with the corresponding component of the reference function in Figure 6.

Matrices in $SO(3)$ induce rotations in the three-dimensional Euclidean space. Such rotations may be visualized by their action on an object. In Figure 7, we compare the Hermite $SO(3)$ -interpolant and the reference $SO(3)$ -function via showing their actions on a tea pot object. Both the interpolant and the reference function are queried at the parameter locations that are marked with a star in the sampling plan displayed in Figure 3. The interpolated rotations visually agree very well with the reference rotations.

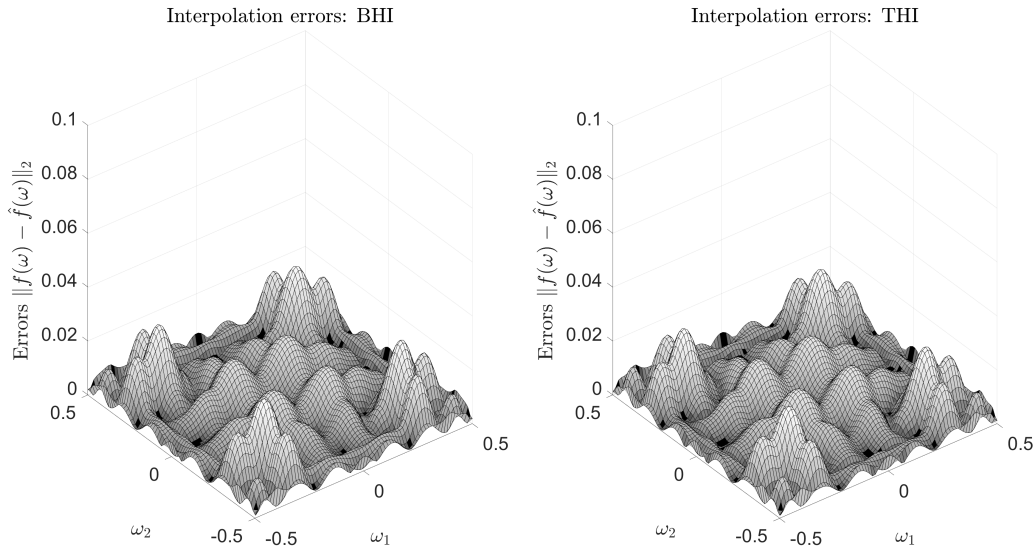


Figure 4: Error surfaces for $SO(3)$ -interpolation on a Chebychev 7×7 grid from Subsection 5.2. Left: Barycentric Hermite Interpolation (BHI). Right: Tangent Space Hermite Interpolation (THI).

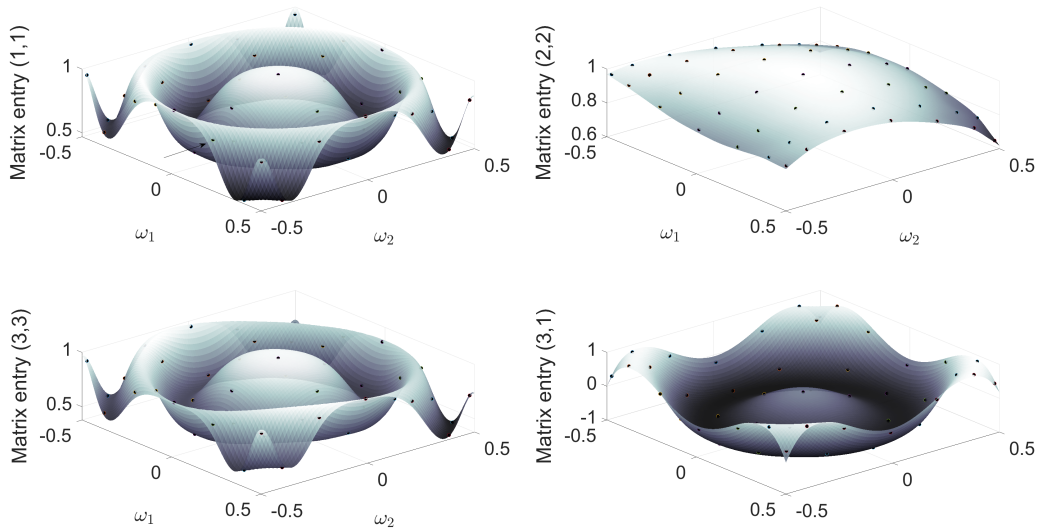


Figure 5: Plots of some selected interpolated matrix component functions $(\omega_1, \omega_2) \rightarrow \left(\hat{f}(\omega_1, \omega_2)\right)_{i,j} \in \mathbb{R}$ of the experiment from Subsection 5.2. The black dots indicate the Chebychev 7×7 sample grid.

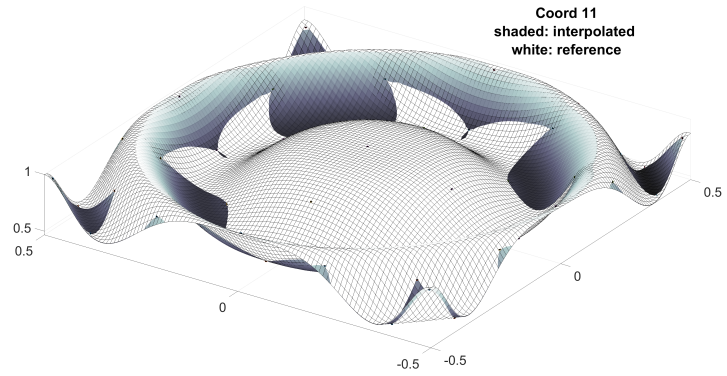


Figure 6: Interpolated matrix component function $\hat{P}_{11} = (\hat{f}(\omega))_{11}$ (shaded surface) and the reference matrix component $P_{11} = f(\omega)$ (white surface) together with the sample locations on a Chebyshev 7×7 grid from Subsection 5.2.

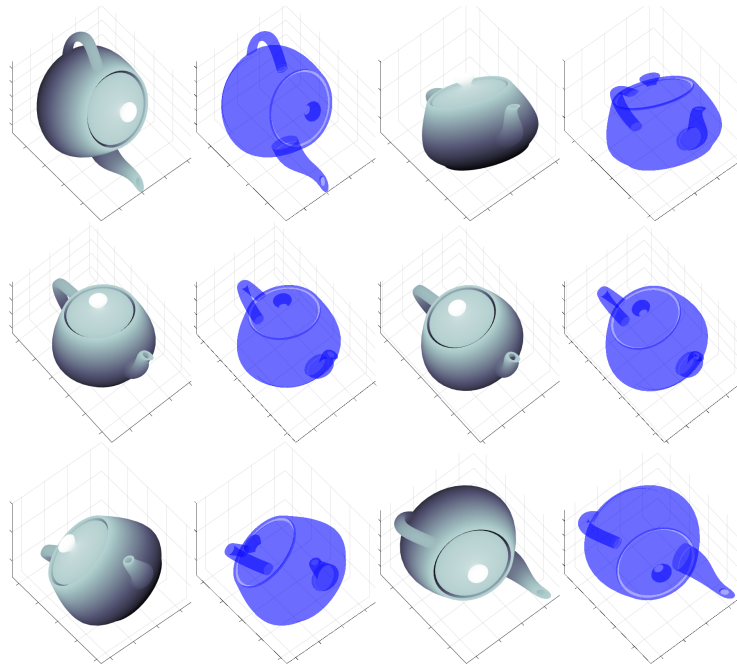


Figure 7: (from upper left to lower right): reference rotations (gray) and interpolated $SO(3)$ -matrices (blue) at the 6 trial points displayed in Fig. 3. The rotation matrices are visualized via their action on the tea pot object.

6. CONCLUSIONS AND DISCUSSION

We have developed two approaches for multivariate Hermite interpolation of manifold-valued data sets. On the one hand, Hermite interpolation can be achieved by computing correspondingly weighted Riemannian barycenters, referred to as *barycentric Hermite interpolation* (BHI). The method we presented should be seen as a generic framework for BHI. For any practical application, the user has to

- construct a sample plan, i.e. employ a method to select the sample points $\omega_j \in \mathbb{R}^d$ (design of experiment).
- choose a Riemannian optimization method to compute the weighted Riemannian center of mass in (2). (Gradient descend, Gauss-Newton-type methods, nonlinear conjugate gradients, etc.)
- choose interpolation weight functions $\omega \rightarrow \varphi_j(\omega)$ with the required properties. Note that the φ_j 's are scalar functions on a Euclidean domain so that this subtask does not require any Riemannian considerations.
- specify side constraints to obtain unique, smooth solutions to the underdetermined linear system in step 5 of Algorithm 2.

On the other hand, Hermite interpolation can be achieved by translating any classical approach of Hermite interpolation from Euclidean vector spaces to the tangent space of the manifold under consideration, referred to as *tangent space Hermite interpolation* (THI). This requires to single out a manifold location and use it as the center for the tangent space. Moreover, all data, i.e., locations and tangent vectors have to be moved to the same tangent space. As a consequence, the interpolant depends on the base-point selection and the preprocessing stage is more involved. Eventually, the interpolant is written as a weighted linear combination of the tangent space images of the sample points and the transported, sampled tangent space vectors. This entails that the number of weight functions in the THI approach equals the number of sample locations plus the total number of sampled partial derivatives. In a sense, THI can be considered a ‘brute-force’ approach. Data is forced into a ‘Procrustean bed’ (=the selected tangent space) and processed without regard to its nature.

From an aesthetic view point, the BHI method is the more appealing one: It is base-point independent and works with a number of weight coefficients that equals the number of sample points. Plus, it treats sample locations and tangent vectors, which in fact are incompatible entities, differently. Yet, in the numerical experiments, the THI method proved to produce the more accurate results and the computational effort of querying the interpolant in the online stage is much lower than for BHI, because the latter involves solving a Riemannian optimization problem at every trial point.

ACKNOWLEDGEMENTS

The original idea for this research paper stems from a short-term visit of the first author to the "Numerical Algorithms and High-Performance Computing"-group at EPF Lausanne in Spring 2022. The first author would like to thank the head of group Daniel Kressner and the doctoral assistant Axel E. J. Séguin for their hospitality and for the very stimulating discussions during that visit.

APPENDIX

A. BARYCENTRIC HERMITE INTERPOLATION: THE EUCLIDEAN CASE

To ease the transition to the manifold case of barycentric interpolation, in this section, we outline multivariate, gradient-enhanced barycentric interpolation in Euclidean spaces.

Let $f: \mathbb{R}^d \rightarrow \mathbb{R}^m$ be differentiable and suppose that we have sampled data $f(w_j) = p_j \in \mathbb{R}^m$, $j = 1, \dots, k$ and partial derivatives $\partial_i f(w_j) = v_j^i \in T_{p_j} \mathbb{R}^m \simeq \mathbb{R}^m$, $j = 1, \dots, k$, $i = 1, \dots, d$. The task is to construct an interpolant \hat{f} such that

$$\hat{f}(w_j) = p_j \in \mathbb{R}^m, \quad j = 1, \dots, k \quad (17)$$

$$\partial_i \hat{f}(w_j) = v_j^i \in T_{p_j} \mathbb{R}^m, \simeq \mathbb{R}^m, \quad j = 1, \dots, k; \quad i = 1, \dots, d. \quad (18)$$

We approach this task by constructing weighted barycenters

$$\hat{f}(\omega) = \arg \min_{q \in \mathbb{R}^m} \frac{1}{2} \sum_{j=1}^k \varphi_j(\omega) \|q - p_j\|_2^2 := \arg \min_{q \in \mathbb{R}^m} L(q, \omega). \quad (19)$$

Here, the φ_j , $j = 1, \dots, k$ are weight functions with the following properties

$$\varphi_j(\omega_i) = \delta_{ij}, \quad \sum_{j=1}^k \varphi_j(\omega) \equiv 1.$$

The objective function $L(q, \omega)$ in (19) is a weighted sum over squared distance terms $\frac{1}{2} \text{dist}(q, p_j)^2 = \frac{1}{2} \|p_j - q\|_2^2 =: L_{p_j}(q)$. The gradient is $\text{grad} L_{p_j}(q) = (q - p_j)$. We emphasize that this is exactly minus the Riemannian logarithm $\text{Log}_q(p_j) = p_j - q$ on the Euclidean \mathbb{R}^m . The minimizer of (19) is the unique zero of the gradient equation

$$0 = \text{grad}_q L(q, \omega) = \sum_{j=1}^k \varphi_j(\omega) \text{grad} L_{p_j}(q) = \sum_{j=1}^k \varphi_j(\omega) (q - p_j). \quad (20)$$

Because of $\sum_{j=1}^k \varphi_j(\omega) \equiv 1$, we obtain the interpolant $q = \hat{f}(\omega)$ as

$$\hat{f}(\omega) = \sum_{j=1}^k \varphi_j(\omega) p_j. \quad (21)$$

Remark A.1. The interpolant in (21) is a weighted linear combination of the sample data points and can thus be constructed in the same fashion in arbitrary vector spaces. Lagrange interpolation, radial basis function interpolation and Gaussian process regression/Kriging eventually boil down to the form of (21) and differ only by the choice of the weight functions $\omega \rightarrow \varphi_j(\omega)$. Hence, all these methods may be considered as examples of weighted barycentric interpolation approaches.

Parameterized solutions via the implicit function theorem [Introduce](#)

$$G: \mathbb{R}^m \times \mathbb{R}^d \rightarrow \mathbb{R}^m, \quad (q, \omega) \mapsto G(q, \omega) = \text{grad}_q L(q, \omega) = \sum_{j=1}^k \varphi_j(\omega)(q - p_j).$$

Suppose that G has a root at (q^*, ω^*) . The differential of G by q at (q^*, ω^*) is

$$D_q G(q^*, \omega^*) = \sum_{j=1}^k \varphi_j(\omega^*) I = I,$$

and is invertible. By the implicit function theorem, there exists $\hat{f}: \mathbb{R}^d \rightarrow \mathbb{R}^m$, $\omega \mapsto \hat{f}(\omega)$ such that on a suitably small domain around ω^* , it holds $G(\hat{f}(\omega), \omega) \equiv 0$. Implicit differentiation yields (locally)

$$D_q G(\hat{f}(\omega), \omega) \circ D\hat{f}(\omega) + D_\omega G(\hat{f}(\omega), \omega) = 0.$$

Note that $D_q G(\hat{f}(\omega), \omega) = D_q \text{grad}_q L(\hat{f}(\omega), \omega) = \text{Hess}_q L(\hat{f}(\omega), \omega)$. Hence, we obtain the following equation that relates the partial derivatives of \hat{f} , which acts as the interpolant, to those of the weight functions φ_j .

$$\text{Hess}_q L(\hat{f}(\omega), \omega) [D\hat{f}(\omega)] = - \sum_{j=1}^k \partial_i \varphi_k(\omega) (\hat{f}(\omega) - p_j) = \sum_{j=1}^k \partial_i \varphi_k(\omega) \text{Log}_{\hat{f}(\omega)}(p_j). \quad (22)$$

It holds $\text{Hess}_q L(\hat{f}(\omega), \omega) = I$ so that

$$D\hat{f}(\omega) = \begin{pmatrix} \text{Log}_{\hat{f}(\omega)}(p_1) & \cdots & \text{Log}_{\hat{f}(\omega)}(p_k) \end{pmatrix} \begin{pmatrix} \partial_i \varphi_1(\omega) \\ \vdots \\ \partial_i \varphi_k(\omega) \end{pmatrix}.$$

If the partial derivatives $\partial_i \hat{f}(\omega_j)$, $i = 1, \dots, d$ are known at the sample sites ω_j , $j = 1, \dots, k$, then the above equation imposes conditions on the partial derivatives of the weight functions thereat. At sample location ω_l , it holds $\hat{f}(\omega_l) = p_l$ and

$$\partial_i \hat{f}(\omega_l) = - \sum_{j=1, j \neq l}^k \partial_i \varphi_j(\omega_l) (p_l - p_j) = \sum_{j=1, j \neq l}^k \partial_i \varphi_j(\omega_l) \text{Log}_{p_l}(p_j). \quad (23)$$

In order to be able to match any prescribed, sampled partial derivatives, $\partial_i \hat{f}(\omega_l)$, we require the sets

$$\mathcal{L}_l := \{p_j - p_l \mid j = 1, \dots, k, j \neq l\} = \{\text{Log}_{p_l}(p_j) \mid j \neq l\} \subset T_{p_l} \mathbb{R}^m \simeq \mathbb{R}^m$$

to span the full tangent space $T_{p_l} \mathbb{R}^m \simeq \mathbb{R}^m$. As a consequence, at least $k \geq m + 1$ sample points must be available to make the barycentric Hermite interpolation problem well-defined. If $k > m + 1$, the equation system (23) is underdetermined and infinitely many valid choices of partial derivative values $\partial_i \varphi_j(\omega_l)$ exist that allow to represent the sampled partial derivatives.

B. KARCHER'S COMPUTATION OF THE GRADIENT AND THE HESSIAN OF THE RIEMANNIAN DISTANCE FUNCTION

To make this paper self-contained, we recap the proof of Karcher's Theorem, [Theorem 2.2](#). To this end, we first list two standard facts about covariant derivatives.

Lemma B.1 (Product rule, do Carmo 1992, Chapter 2, Prop. 3.2). *Let $\frac{D}{dt}$ denote the Levi-Civita connection along a curve $c : I \rightarrow \mathcal{M}$. It holds*

$$\frac{d}{dt} \langle X(t), Y(t) \rangle_{c(t)} = \left\langle \frac{DX}{dt}(t), Y(t) \right\rangle_{c(t)} + \left\langle X(t), \frac{DY}{dt}(t) \right\rangle_{c(t)}$$

for all smooth vector fields X, Y along c .

The next statement corresponds to the Theorem of Schwarz in classical multivariate calculus.

Lemma B.2 (do Carmo 1992, Chapter 3, Lemma 3.4). *Let $\alpha : I \times J \rightarrow \mathcal{M}, (s, t) \mapsto \alpha(s, t)$ be smooth. Note that for s fixed, $t \mapsto \alpha(s, t)$ is a smooth manifold curve, likewise for t fixed. Hence,*

$$\begin{aligned} \frac{\partial \alpha}{\partial s}(s, t) &= \left. \frac{d}{d\sigma} \right|_{\sigma=0} \alpha(s + \sigma, t) \in T_{\alpha(s,t)} \mathcal{M}, \\ \frac{\partial \alpha}{\partial t}(s, t) &= \left. \frac{d}{d\tau} \right|_{\tau=0} \alpha(s, t + \tau) \in T_{\alpha(s,t)} \mathcal{M}, \end{aligned}$$

and it holds

$$\frac{D}{dt} \frac{\partial \alpha}{\partial s}(s, t) = \frac{D}{ds} \frac{\partial \alpha}{\partial t}(s, t).$$

Here,

$$\frac{D}{dt} \frac{\partial \alpha}{\partial s}(s, t) := \left. \frac{D}{d\tau} \right|_{\tau=0} \tau \mapsto \alpha(s, t + \tau).$$

Likewise for $\frac{D}{ds} \frac{\partial \alpha}{\partial t}(s, t)$.

Theorem B.3 (Karcher 1977, Thm. 1.2). *Let \mathcal{M} be a complete Riemannian manifold and let $p \in \mathcal{M}$ be a fixed point. Let $B_\rho(p)$ be a geodesic ball around p_* such that the geodesics between any two points inside $B_\rho(p)$ are unique and minimizing. Define*

$$f : B_\rho(p) \rightarrow \mathbb{R}, \quad q \mapsto \frac{1}{2} \text{dist}(q, p)^2.$$

Then

$$\text{grad}_q f = -\text{Log}_q(p).$$

Proof. Let $\gamma : [0, 1] \rightarrow B_\rho(p)$ be a geodesic with $\gamma(0) = q, \dot{\gamma}(0) = v \in T_q \mathcal{M}$. Consider a variation of geodesics

$$c_p(s, t) = \text{Exp}_p(s \text{Log}_p(\gamma(t)))$$

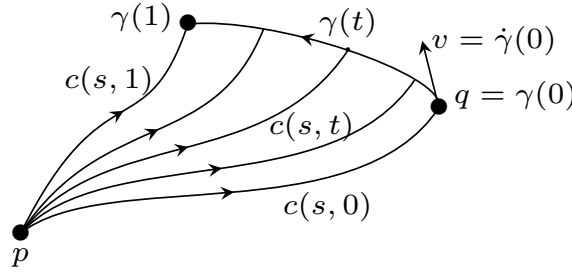


Figure 8: Illustration of the variation through geodesics considered in the proof of Theorem B.3.

as illustrated in Figure 8. As in Karcher 1977, we write

$$c'_p(s, t) = \partial_s c_p(s, t), \quad \dot{c}_p(s, t) = \partial_t c_p(s, t)$$

and observe that

- (i) $c_p(0, t) = \text{Exp}_p(0) = p$. (All geodesics $s \mapsto c_p(s, t)$ start from p).
- (ii) $c_p(1, t) = \gamma(t)$. (All geodesics $s \mapsto c_p(s, t)$ end at $\gamma(t)$). Moreover, $\dot{\gamma}(t) = \dot{c}_p(1, t)$.
- (iii) the terminal velocity of the geodesic $s \mapsto c_p(s, t)$ matches minus the velocity of the geodesic emanating from $\gamma(t) = c_p(1, t)$ to reach p . Hence,

$$c'_p(1, t) = -\text{Log}_{\gamma(t)}(p).$$

- (iv) all geodesics are constant-speed curves so that

$$\text{dist}(p, \gamma(t)) = \|\text{Log}_p(\gamma(t))\|_p = \|c'_p(0, t)\|_p = \|c'_p(s, t)\|_{c_p(s, t)} \text{ for all } s \in [0, 1].$$

Because of this, we can write

$$\|c'_p(s, t)\|^2 = \int_0^1 \|c'_p(s, t)\|^2 ds.$$

- (v) When taking covariant derivatives, we may change the order of differentiation according to Theorem B.2

$$\frac{D}{ds} \dot{c}_p(s, t) = \frac{D}{dt} c'_p(s, t).$$

We calculate for any $v \in T_q \mathcal{M}$ the differential as

$$\begin{aligned}
 df_q(v) &= \left. \frac{d}{dt} f(\gamma(t)) \right|_{t=0} = \left. \frac{1}{2} \frac{d}{dt} \text{dist}(\gamma(t), p)^2 \right|_{t=0} \\
 &= \left. \frac{1}{2} \frac{d}{dt} \int_0^1 \|c'_p(s, t)\|^2 ds \right|_{t=0} \\
 &= \left. \frac{1}{2} \int_0^1 \frac{d}{dt} \langle c'_p(s, t), c'_p(s, t) \rangle ds \right|_{t=0} \\
 &= \int_0^1 \left. \left\langle \frac{D}{dt} c'_p(s, t), c'_p(s, t) \right\rangle \right|_{t=0} ds \\
 &= \int_0^1 \left. \left\langle \frac{D}{ds} \dot{c}_p(s, t), c'_p(s, t) \right\rangle \right|_{t=0} ds \\
 &= \int_0^1 \left(\left. \left\langle \frac{D}{ds} \dot{c}_p(s, t), c'_p(s, t) \right\rangle + \langle \dot{c}_p(s, t), \frac{D}{ds} c'_p(s, t) \rangle \right) \right|_{t=0} ds,
 \end{aligned}$$

where the second term vanishes, since $c_p(s, t)$ is a geodesic in s for any fixed t . We continue

$$\begin{aligned}
 df_q(v) &= \int_0^1 \left. \frac{d}{ds} \langle \dot{c}_p(s, t), c'_p(s, t) \rangle ds \right|_{t=0} \\
 &= \left[\langle \dot{c}_p(s, t), c'_p(s, t) \rangle \right]_0^1 \Big|_{t=0}, \\
 &= \langle \dot{c}_p(1, t), c'_p(1, t) \rangle + \langle \dot{c}_p(0, t), c'_p(0, t) \rangle \Big|_{t=0}, \tag{24}
 \end{aligned}$$

where again the second summand is zero since $c_p(0, t) = p$ is constant. We finally obtain

$$df_q(v) = \langle \dot{\gamma}(t), -\text{Log}_{\gamma(t)}(p) \rangle \Big|_{t=0} = \langle v, -\text{Log}_q(p) \rangle$$

and hence $\text{grad}_q f = -\text{Log}_q(p) \in T_q \mathcal{M}$. □

The Hessian of the Riemannian distance function. Let $g: \mathcal{M} \rightarrow \mathbb{R}$ be a smooth scalar function. Following Kühnel 2015, Section 6A, it may be considered as a $(0, 0)$ -tensor. The covariant derivative of g (in the sense of Kühnel 2015, Def. 6.2) is a $(0, 1)$ -tensor $(\nabla g)(X)$ that maps a vector field X to a scalar function. More precisely, we have

$$\nabla g: \mathcal{X}(\mathcal{M}) \rightarrow C^\infty(\mathcal{M}), X \mapsto (p \mapsto (\nabla f(X))(p) = dg_p(X(p)) = \langle \text{grad } f(p), X(p) \rangle_p).$$

The second covariant derivative, $\nabla^2 g$, called the *Hessian* of g , is obtained by taking the covariant derivative of the $(0, 1)$ -tensor (∇g) . This produces a $(0, 2)$ -tensor. A calculation shows

$$\nabla^2 f(X, Y) = \langle \nabla_X \text{grad } f, Y \rangle.$$

The Hessian $(1, 1)$ -tensor associated with g is thus

$$\text{Hess } f: \mathcal{X}(\mathcal{M}) \rightarrow \mathcal{X}(\mathcal{M}), \quad X \mapsto \nabla_X \text{grad } g.$$

The next lemma provides a path to compute the Hessian operator at a point p and tangent vector $v = X(p) \in T_p \mathcal{M}$.

Lemma B.4 (cf. Absil, Mahony, and Sepulchre 2008, Prop. 5.5.4). *Let $g: \mathcal{M} \rightarrow \mathbb{R}$ be a smooth scalar function on a Riemannian manifold \mathcal{M} . Let γ be a geodesic with $\gamma(0) = q$, $\dot{\gamma}(0) = v$. Then*

$$\langle \text{Hess } g(q)[v], v \rangle = \frac{d^2}{dt^2} \Big|_{t=0} (g \circ \gamma)(t).$$

Proof. It holds

$$\frac{d}{dt} (f \circ \gamma)(t) = df_{\gamma(t)}(\dot{\gamma}(t)) = \langle \text{grad } f(\gamma(t)), \dot{\gamma}(t) \rangle.$$

Taking the second derivative, we obtain by the rules of covariant differentiation along a curve and the product rule of Lemma B.1 that

$$\begin{aligned} \frac{d^2}{dt^2} (f \circ \gamma)(t) &= \frac{d}{dt} \langle \text{grad } f(\gamma(t)), \dot{\gamma}(t) \rangle \\ &= \left\langle \frac{D}{dt} \text{grad } f(\gamma(t)), \dot{\gamma}(t) \right\rangle + \langle \text{grad } f(\gamma(t)), \frac{D}{dt} \dot{\gamma}(t) \rangle \\ &= \langle \nabla_{\dot{\gamma}(t)} \text{grad } f, \dot{\gamma}(t) \rangle \\ &= \langle \text{Hess } f(\gamma(t))[\dot{\gamma}(t)], \dot{\gamma}(t) \rangle. \end{aligned}$$

The third identity holds, because $\frac{D}{dt} \dot{\gamma}(t) = 0$, since γ is a geodesic. At $t = 0$, this yields

$$\frac{d^2}{dt^2} (f \circ \gamma)(0) = \langle \text{Hess } f(q)[v], v \rangle, \quad (25)$$

which finishes the proof. \square

For computing the Hessian of the distance function $q \mapsto \frac{1}{2} \text{dist}(p, q)^2$, we utilize the same variation of geodesics $c_p(s, t)$ as in the above proof of Theorem B.3 and continue the calculation from (24). Using that $c_p(1, t)$ is a geodesic, we obtain

$$\begin{aligned} \frac{d^2}{dt^2} \frac{1}{2} \text{dist}(\gamma(t), p)^2 &= \frac{d}{dt} \left(\frac{d}{dt} \frac{1}{2} \text{dist}(\gamma(t), p)^2 \right) \stackrel{(24)}{=} \frac{d}{dt} \langle \dot{c}_p(1, t), c'_p(1, t) \rangle \\ &= \left\langle \frac{D}{dt} \dot{c}_p(1, t), c'_p(1, t) \right\rangle + \left\langle \dot{c}_p(1, t), \frac{D}{dt} c'_p(1, t) \right\rangle \\ &= \left\langle \dot{\gamma}(t), \frac{D}{ds} \dot{c}_p(1, t) \right\rangle. \end{aligned}$$

At $t = 0$, we obtain for $\gamma(0) = q$, $\dot{\gamma}(0) = v$,

$$\langle \text{Hess } f(q)[v], v \rangle = \frac{d^2}{dt^2} \Big|_{t=0} \text{dist}(\gamma(t), p) = \left\langle \frac{D}{ds} \dot{c}_p(1, 0), v \right\rangle. \quad (26)$$

Remark Associated with the variation $c_p(s, t) = \text{Exp}_p(s \text{Log}_p(\gamma(t)))$ is the variation field

$$s \mapsto \dot{c}_p(s, 0) = \partial_t \Big|_{t=0} c_p(s, t) = d(\text{Exp}_p)_{(s \text{Log}_p(q))} \left(s \cdot d(\text{Log}_p)_q(v) \right)$$

For notational convenience, define $x = \text{Log}_p(q)$, $y = d(\text{Log}_p)_q(v) \in T_p \mathcal{M}$. The above variation vector field coincides with

$$s \mapsto \dot{c}_p(s, 0) = \partial_t \Big|_{t=0} c_p(s, t) = \partial_\tau \Big|_{\tau=0} \text{Exp}_p(s \cdot (x + \tau y)).$$

This is a variation of the geodesic $s \mapsto \hat{c}_p(s) := c_p(s, 0)$ from p to q through geodesics, thus a Jacobi field.⁴ Hence, with $s \mapsto J(s) = \partial_\tau|_{\tau=0} \text{Exp}_p(s \cdot (x + \tau y))$, we have

$$J(s) = \dot{c}_p(s, 0), \quad J'(s) = \frac{D}{ds} J(s) = \frac{D}{ds} \dot{c}_p(s, 0).$$

Thus, we may write

$$\langle \text{Hess } f(q)[v], v \rangle = \langle J'(1), v \rangle.$$

REFERENCES

- Absil, P.-A., P.-Y. Gouenbourger, P. Striowski, and B. Wirth (2016). “Differentiable Piecewise-Bézier Surfaces on Riemannian Manifolds”. In: *SIAM Journal on Imaging Sciences* 9.4, pp. 1788–1828. URL: [10.1137/16M1057978](https://doi.org/10.1137/16M1057978).
- Absil, P.-A., R. Mahony, and R. Sepulchre (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press. DOI: [10.1515/9781400830244](https://doi.org/10.1515/9781400830244).
- Afsari, B., R. Tron, and R. Vidal (2013). “On the convergence of gradient descent for finding the Riemannian center of mass”. In: *SIAM Journal on Control and Optimization* 51.3, pp. 2230–2260. DOI: [10.1137/12086282X](https://doi.org/10.1137/12086282X).
- Allasia, G., C. Roberto, and A. De Rossi (2018). “Hermite–Birkhoff interpolation on scattered data on the sphere and other manifolds”. In: *Applied Mathematics and Computation* 318. Recent Trends in Numerical Computations: Theory and Algorithms, pp. 35–50. DOI: [10.1016/j.amc.2017.05.018](https://doi.org/10.1016/j.amc.2017.05.018).
- Amsellem, D. (2010). “Interpolation on Manifolds of CFD-based Fluid and Finite Element-based Structural Reduced-order Models for On-line Aeroelastic Prediction”. PhD thesis. Stanford University.
- Bergmann, R. and P.-Y. Gouenbourger (2018). “A variational model for data fitting on manifolds by minimizing the acceleration of a Bézier curve”. In: *Frontiers in Applied Mathematics and Statistics*. DOI: [10.3389/fams.2018.00059](https://doi.org/10.3389/fams.2018.00059). arXiv: [1807.10090](https://arxiv.org/abs/1807.10090).
- Buhmann, M. D. (2003). *Radial Basis Functions*. Vol. 12. Cambridge Monographs on Applied and Computational Mathematics. Cambridge, UK: Cambridge University Press.
- Do Carmo, M. P. (1992). *Riemannian Geometry*. Mathematics: Theory & Applications. Birkhäuser Boston, Inc., Boston, MA.
- Dyn, N. and N. Sharon (2017). “Manifold-valued subdivision schemes based on geodesic inductive averaging”. In: *Journal of Computational and Applied Mathematics* 311, pp. 54–67. DOI: [10.1016/j.cam.2016.07.008](https://doi.org/10.1016/j.cam.2016.07.008).
- Forrester, A. I. J., A. Sobester, and A. J. Keane (2008). *Engineering Design via Surrogate Modelling: A Practical Guide*. United Kingdom: John Wiley & Sons.
- Golub, G. . and C. F. Van Loan (1996). *Matrix Computations*. 3rd. Baltimore – London: The John Hopkins University Press.

⁴More precisely, the unique Jacobi field along the geodesic $c_p(s, 0)$ that matches the initial values $J(0) = \partial_\tau|_{\tau=0} \text{Exp}_p(0 \cdot (x + \tau y)) = \partial_\tau|_{\tau=0} p = 0$ and

$$\begin{aligned} J'(0) &= \frac{D}{ds} \Big|_{s=0} \partial_\tau \Big|_{\tau=0} \text{Exp}_p(s \cdot (x + \tau y)) = \frac{D}{d\tau} \Big|_{\tau=0} \partial_s \Big|_{s=0} \text{Exp}_p(s \cdot (x + \tau y)) \\ &= \frac{D}{d\tau} \Big|_{\tau=0} (x + \tau y) = y = d(\text{Log}_p)_q(v). \end{aligned}$$

- Gousenbourger, P.-Y., E. Massart, and P.-A. Absil (2018). “Data Fitting on Manifolds with Composite Bézier-Like Curves and Blended Cubic Splines”. In: *Journal of Mathematical Imaging and Vision* online, pp. 1–27. DOI: [10.1007/s10851-018-0865-2](https://doi.org/10.1007/s10851-018-0865-2).
- Grohs, P., H. Hardering, and O. Sander (2015). “Optimal a priori discretization error bounds for geodesic finite elements”. In: *Foundations of Computational Mathematics* 15.6, pp. 1357–1411. DOI: [10.1007/s10208-014-9230-z](https://doi.org/10.1007/s10208-014-9230-z).
- Higham, N. J. (2008). *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, pp. xx+425.
- Jakubiak, J., F. S. Leite, and R. Rodrigues (2006). “A two-step algorithm of smooth spline generation on Riemannian manifolds”. In: *Journal of Computational and Applied Mathematics* 194, pp. 177–191. DOI: [10.1016/j.cam.2005.07.003](https://doi.org/10.1016/j.cam.2005.07.003).
- Karcher, H. (1977). “Riemannian center of mass and mollifier smoothing”. In: *Communications on Pure and Applied Mathematics* 30.5, pp. 509–541. DOI: [10.1002/cpa.3160300502](https://doi.org/10.1002/cpa.3160300502).
- Kühnel, W. (2015). *Differential Geometry: Curves – Surfaces – Manifolds*. 3rd ed. Student Mathematical Library. American Mathematical Society.
- Lee, J. M. (2012). *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer New York.
- Lee, J. M. (1997). *Riemannian Manifolds*. Vol. 176. Graduate Texts in Mathematics. An introduction to curvature. Springer-Verlag, New York. DOI: [10.1007/b98852](https://doi.org/10.1007/b98852).
- Narcowich, F. (1995). “Generalized Hermite Interpolation and Positive Definite Kernels on a Riemannian Manifold”. In: *Journal of Mathematical Analysis and Applications* 190, pp. 165–193.
- Nava-Yazdani, E. and K. Polthier (2013). “De Casteljau’s algorithm on manifolds”. In: *Computer Aided Geometric Design* 30.7, pp. 722–732. DOI: [10.1016/j.cagd.2013.06.002](https://doi.org/10.1016/j.cagd.2013.06.002).
- Popiel, T. and L. Noakes (2007). “Bézier curves and C_2 interpolation in Riemannian manifolds”. In: *Journal of Approximation Theory* 148.2, pp. 111–127.
- Samir, C. and I. Adouani (2019). “ C_1 interpolating Bézier path on Riemannian manifolds, with applications to 3D shape space”. In: *Applied Mathematics and Computation* 348, pp. 371–384. DOI: [10.1016/j.amc.2018.11.060](https://doi.org/10.1016/j.amc.2018.11.060).
- Sander, O. (2012). “Geodesic finite elements on simplicial grids”. In: *International journal for numerical methods in engineering* 92.12, pp. 999–1025.
- Sander, O. (2016). “Geodesic finite elements of higher order”. In: *IMA Journal of Numerical Analysis* 36 (1), pp. 238–266. DOI: [10.1093/imanum/drv016](https://doi.org/10.1093/imanum/drv016).
- Séguin, A. and D. Kressner (2022). “Continuation Methods for Riemannian Optimization”. In: *SIAM Journal on Optimization* 32.2, pp. 1069–1093. DOI: [10.1137/21M1428650](https://doi.org/10.1137/21M1428650). arXiv: [2106.08839](https://arxiv.org/abs/2106.08839).
- Vardi, H. B.-Z., N. Dyn, and N. Sharon (2022). *Geometric Hermite Interpolation in \mathbb{R}^n by Refinements*. arXiv: [2203.02903](https://arxiv.org/abs/2203.02903).
- Wallner, J. and N. Dyn (2005). “Convergence and C_1 analysis of subdivision schemes on manifolds by proximity”. In: *Computer Aided Geometric Design* 22.7. Geometric Modelling and Differential Geometry, pp. 593–622. DOI: [10.1016/j.cagd.2005.06.003](https://doi.org/10.1016/j.cagd.2005.06.003).
- Zimmermann, R. (2013). “On the Maximum Likelihood Training of Gradient-Enhanced Spatial Gaussian Processes”. In: *SIAM Journal on Scientific Computing* 35.6, A2554–A2574. DOI: [10.1137/13092229X](https://doi.org/10.1137/13092229X).
- Zimmermann, R. (2020). “Hermite Interpolation and Data Processing Errors on Riemannian Matrix Manifolds”. In: *SIAM Journal on Scientific Computing* 42.5, A2593–A2619. DOI: [10.1137/19M1282878](https://doi.org/10.1137/19M1282878).
- Zimmermann, R. (2021). “Manifold interpolation”. In: *System- and Data-Driven Methods and Algorithms*. Ed. by P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira. Vol. 1. Model Order Reduction. Boston: De Gruyter, pp. 229–274.