Differentiable piecewise-Bézier surfaces on Riemannian manifolds

P.-A. Absil* Pierre-Yves Gousenbourger*
Paul Striewski† Benedikt Wirth†

25th November 2015

Abstract

We generalize the notion of Bézier surfaces and surface splines to Riemannian manifolds. To this end we put forward and compare three possible alternative definitions of Bézier surfaces. We furthermore investigate how to achieve $\mathcal{C}^0$- and $\mathcal{C}^1$-continuity of Bézier surface splines. Unlike in Euclidean space and for one-dimensional Bézier splines on manifolds, $\mathcal{C}^1$-continuity cannot be ensured by simple conditions on the Bézier control points: it requires an adaptation of the Bézier spline evaluation scheme. Finally, we propose an algorithm to optimize the Bézier control points given a set of points to be interpolated by a Bézier surface spline. We show computational examples on the sphere, the special orthogonal group and two Riemannian shape spaces.

Keywords: Composite Bézier surface, Riemannian manifold, differentiability conditions, bending energy.

Mathematics Subject Classification (2010): 65D05, 65D07, 53C22, 65K10

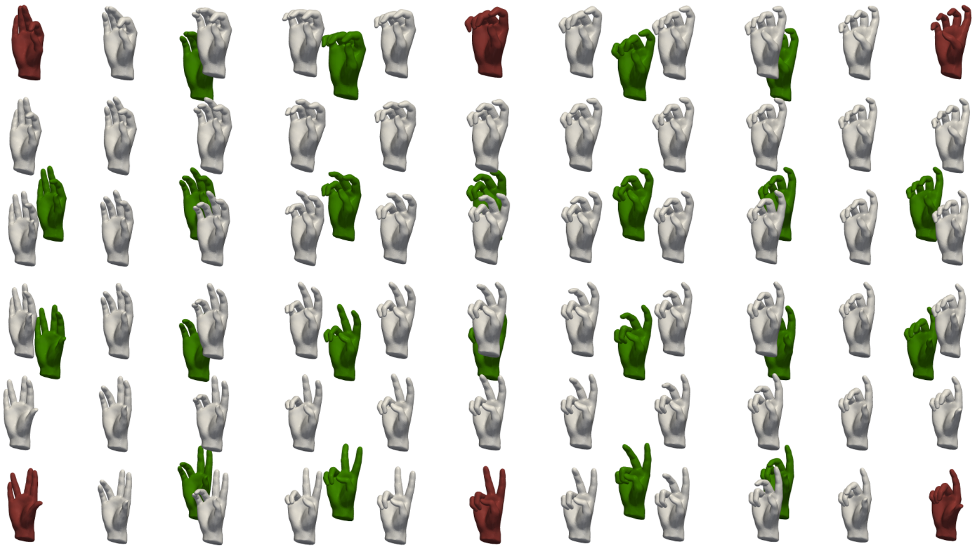


Figure 1: Differentiable Bézier spline surface on the Riemannian space of shells (Section 6.4) interpolating the red shapes. The gray shapes are points on the Bézier surface driven by the control points in green. Their location indicates where in the $\mathbb{R}^2$ domain they are achieved.

*ICTEAM Institute, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

†Institute for Numerical and Applied Mathematics, University of Münster, Einsteinstraße 62, D-48149 Münster, Germany

1 Introduction

During recent years, it has become more and more common and important to process data from non-Euclidean spaces, in particular from Riemannian manifolds. Examples for the use of manifold-valued data include the exploration of Riemannian shape spaces in computer vision [9], the interpretation of colors in images as data on the color circle $\mathbb{S}^1$ [3], the representation of fixed rank matrices as a submanifold of all matrices [31], and many more.

This paper concerns multivariate manifold-valued interpolation. More precisely, given data points $p_{i_1,\ldots,i_d}$ on a manifold $\mathcal{M}$ associated to nodes $(i_1,\ldots,i_d) \in \mathbb{Z}^d$ of a Cartesian grid in $\mathbb{R}^d$, we seek a smooth (*i.e.*, $\mathcal{C}^1$) function $\mathfrak{B} : \mathbb{R}^d \to \mathcal{M}$ such that $\mathfrak{B}(i_1,\ldots,i_d) = p_{i_1,\ldots,i_d}$.

This problem is motivated by several applications in engineering and the sciences. In projection-based model order reduction of a dynamical system that depends on d parameters, a possible approach consists in computing a suitable projector (hence, an element of a Grassmann manifold) for values of the parameters falling on a grid, then resorting to interpolation in order to generate a projector for other values of the parameters [21]. In diffusion tensor imaging, a diffusion tensor (hence, an element of the manifold of 3×3 symmetric positive definite matrices) is acquired for each voxel of the volume of interest (thus $d = 3$), and interpolation can be used to infer more finely sampled data [19]. In Cosserat rods [25], we have $d = 1$ and $\mathcal{M} = \mathrm{SE}(3)$—the group of rigid-body motions in $\mathbb{R}^3$—while Cosserat shells [26] require $d = 2$. Let us finally mention liquid crystals, which can be described by a function from $\mathbb{R}^3$ into the projective space $\mathbb{RP}^2$ [18].

Two special cases of manifold-valued interpolation have been widely considered in the literature. One of them is when $\mathcal{M}$ is a linear manifold, *i.e.*, a Euclidean space. A preferred way of handling Euclidean interpolation problems is by resorting to piecewise-polynomial (or *spline*) interpolation functions. Indeed, polynomial functions are convenient to manipulate and evaluate, and the piecewise approach makes it possible to keep interpolation errors small while using polynomial pieces of low degree, avoiding the problem of Runge's phenomenon that plagues high-degree polynomial interpolation. Within the realm of piecewise-polynomial interpolation, countless variations occur according to the sought degree of smoothness, restrictions on the admissible classes of polynomial pieces, optimality criteria, and also the form—one of them being the Bézier form—in which the polynomials are expressed. The book of Farin [8] is a convenient point of entry to this vast literature.

The other special case of manifold-valued interpolation is when $d = 1$. The interpolation problem then reduces to interpolating time-labeled data points on $\mathcal{M}$ by a curve that goes through these points at the prescribed instants. This problem has received a fair amount of attention in the literature. Recent contributions can be found in [24, 13, 30, 2].

Besides these two special—but important—cases, multivariate manifold-valued interpolation does not appear to have been much researched, in spite of the above-mentioned applications. Steinke *et al.* [28] use a thin-plate-spline technique to produce an interpolation map between two Riemannian manifolds. The approach is generalized in [29] where, given a set of training pairs (X_i, Y_i) with the X_i's and Y_i's on two manifolds, a mapping is sought between these two manifolds that minimizes a regularized empirical risk. We also mention a related technique for volumetric registration presented in [14].

In this paper, we present a technique to perform multivariate manifold-valued interpolation by means of $\mathcal{C}^1$ piecewise-cubic-Bézier functions (see Figure 1 for an example). We thereby extend to manifold codomains the above-mentioned benefits of piecewise-polynomial interpolation. When $d = 1$, the proposed method reduces to a strategy very close to the one developed in [2, 11]. The only difference is that in these articles the two extreme segments are chosen to be quadratic Bézier curves and not cubic Bézier curves. The proposed method can thus almost be viewed as a multivariate extension of [2, §2]. For simplicity of the exposition, we focus on the bivariate case ($d = 2$), but the transition to higher values of d appears to be considerably less intricate than the transition from $d = 1$ to $d = 2$.

The development of the proposed bivariate manifold-valued technique requires more work than one might anticipate. The first task is to propose a bivariate extension of the well-known [17, 20] manifold-valued Bézier curves that constitute the pieces in [2]. We provide three possibilities leading in general to different results (see Section 3, Definition 4), in which the essential component is a weighted geodesic average.

Next comes the question of gluing these Bézier patches together in a smooth ($\mathcal{C}^1$) way. The problem is substantially more difficult than in [2] because the interfaces are no longer isolated points but instead regions of dimension 1. The conditions for $\mathcal{C}^0$-continuity of multiple Bézier surfaces patched together are the same as in Euclidean space, namely that the control points of adjacent patches coincide at the interface; however, the classical conditions on the control points for $\mathcal{C}^1$-continuity in Euclidean space exhibit a linear dependence which turns into an incompatibility on manifolds. As a result, the classical conditions can in general not be satisfied. In addition, those conditions are not sufficient to ensure $\mathcal{C}^1$-continuity on a manifold. We overcome this difficulty by "discarding" the control points that lie on the interfaces. We show in Theorem 18 that $\mathcal{C}^1$ gluing can then be achieved for two of the three Bézier surface definitions.

For the interpolation problem, we provide a technique to fix the remaining leeway on the control points in such a way that the resulting interpolation function has minimal mean squared second derivative when the codomain $\mathcal{M}$ reduces to a Euclidean space. The outcome is then an advanced method to address the bivariate manifold-valued interpolation problem, whose sophistication makes it a promising tool towards making the most of expensively acquired data in applications such as those mentioned above.

The paper is organized as follows. After a reminder in Section 2 on the Euclidean case, piecewise-Bézier curves and surfaces on manifolds are introduced and analyzed in Section 3. The interpolation technique is presented in Section 4. We briefly introduce all necessary numerical tools in Section 5, and in Section 6 we show some numerical applications in the context of motion modeling and shape exploration in shape spaces. Conclusions are drawn in Section 7. For the reader's convenience, a glossary of notations is given at the very end of the article.

2 Reminder on Euclidean piecewise-Bézier curves and surfaces

This section is a brief summary of Bézier curves and surfaces in the Euclidean space $\mathbb{R}^n$. A detailed exposition can be found in [8]. For a sequence of control points $b_0, \dots, b_K \in \mathbb{R}^n$, the corresponding Bézier curve $\beta_K(\cdot; b_0, \dots, b_K) : [0,1] \to \mathbb{R}^n$ is defined as

$$\beta_K(t; b_0, \dots, b_K) = \sum_{j=0}^{K} b_j B_{jK}(t) , \tag{1}$$

where K is called the order of the curve ($K = 3$ for cubic Bézier curves) and $B_{jK}(t)$ denotes the j^{th} Bernstein polynomial of degree K,

$$B_{jK}(t) = \binom{K}{j} t^j (1-t)^{K-j}. \tag{2}$$

The Bézier curve fully lies inside the convex hull of its control points $b_0, \dots, b_K$, since for each fixed t, the $B_{jK}(t)$ form a partition of unity and thus can be interpreted as convex combination coefficients (Figure 2 left). The curve interpolates the first and last control points, and its initial and final velocity are tangent to the initial and final line segments between the control points (Figure 2),

$$\begin{aligned} \beta_K(0; b_0, \dots, b_K) &= b_0 , & \dot{\beta}_K(0; b_0, \dots, b_K) &= K(b_1 - b_0), \\ \beta_K(1; b_0, \dots, b_K) &= b_K , & \dot{\beta}_K(1; b_0, \dots, b_K) &= K(b_K - b_{K-1}), \end{aligned}$$

where $\dot{\beta}$ denotes the time-derivative of β.

De Casteljau's algorithm offers a method to evaluate $\mathfrak{b}_0^K = \beta_K(t_0; b_0, \dots, b_K)$ at a time $t_0 \in [0,1]$,

$$\begin{aligned} \mathfrak{b}_j^0 &= b_j, & j &= 0, \dots, K, \\ \mathfrak{b}_j^k &= t_0\ \mathfrak{b}_{j+1}^{k-1} + (1-t_0)\ \mathfrak{b}_j^{k-1}, & k &= 1, \dots, K,\ j = 0, \dots, K-k. \end{aligned} \tag{3}$$

Note that the operations employed above are only convex combinations with coefficients t_0 and $1-t_0$ so that the algorithm has a simple geometric interpretation (Figure 2 right).

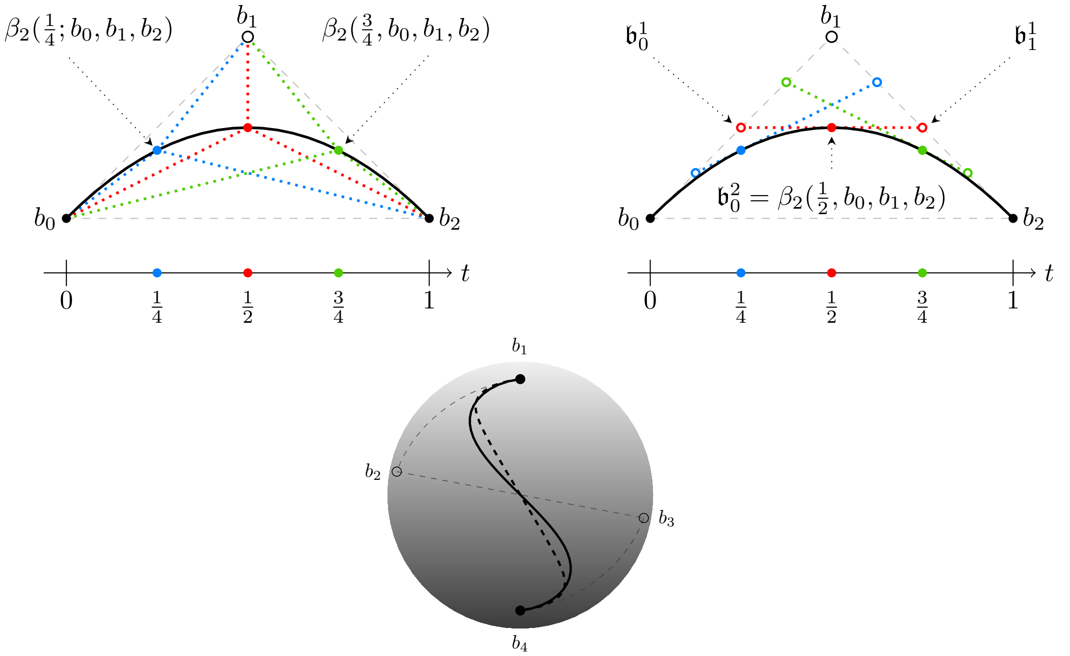


Figure 2: Computation of a quadratic ($K = 2$) Bézier curve via equation (1) as a weighted mean of all control points (top left) and via de Casteljau's algorithm (3) as a recursively iterated weighted mean of point pairs (top right). In Euclidean space both methods are equivalent, but they will usually not lead to identical results if they are generalized to manifolds as the sphere (bottom, showing a cubic Bézier curve). The dashed curve is obtained with the weighted mean and the solid one with de Casteljau's algorithm. Interpolation and control points are indicated by filled and open circles, respectively; the Bézier polygon is shown as gray dashed lines.

Several Bézier curves can be joined to form a Bézier spline: for two Bézier curves $\beta_K(t; b_0^{\mathrm{l}}, \dots, b_K^{\mathrm{l}}), \beta_K(t; b_0^{\mathrm{r}}, \dots, b_K^{\mathrm{r}}) : [0,1] \to \mathbb{R}^n$, the composite spline curve

$$\mathfrak{B} : [0,2] \to \mathbb{R}^n : \quad t \mapsto \begin{cases} \beta_K(t; b_0^{\mathrm{l}}, \dots, b_K^{\mathrm{l}}) & \text{if } t \in [0,1] \\ \beta_K(t-1; b_0^{\mathrm{r}}, \dots, b_K^{\mathrm{r}}) & \text{if } t \in (1,2] \end{cases} \tag{4}$$

is continuous if and only if $b_0^{\mathrm{r}} = b_K^{\mathrm{l}}$ and first order differentiable if and only if $b_K^{\mathrm{l}} = \frac{b_{K-1}^{\mathrm{l}} + b_1^{\mathrm{r}}}{2}$.

The idea of Bézier curves extends also to higher dimensions; see, *e.g.*, [8, Sec. 5.5]. For a family of points $(b_{ij})_{i,j=0,\dots,K} \subset \mathbb{R}^n$, the corresponding Bézier surface $\beta_K(t_1, t_2; (b_{ij})_{i,j=0,\dots,K}) : [0,1]^2 \to \mathbb{R}^n$ is defined by

$$\beta_K(t_1, t_2; (b_{ij})_{i,j=0,\dots,K}) = \sum_{i=0}^{K} \sum_{j=0}^{K} b_{ij} B_{iK}(t_1) B_{jK}(t_2)\,. \tag{5}$$

Again, for fixed t_1 and t_2 this can be interpreted as a convex combination of all control points b_{ij} (Figure 3 left). We directly see that the Bézier surface boundary consists of the four Bézier curves with control points $b_{0,j}$, $b_{K,j}$, $b_{j,0}$, $b_{j,K}$, $j = 0, \dots, K$, respectively. We can also interpret a Bézier surface as a one-parameter family of Bézier curves,

$$\begin{aligned}\beta_K(t_1, t_2; (b_{ij})_{i,j=0,\dots,K}) &= \sum_{j=1}^{K} \left(\sum_{i=1}^{K} b_{ij} B_{iK}(t_1) \right) B_{jK}(t_2) \\ &= \beta_K(t_2; (\beta_K(t_1; (b_{ij})_{i=0,\dots,K}))_{j=0,\dots,K}), \end{aligned} \tag{6}$$

which allows an evaluation just based on the computation of Bézier curves (Figure 3, middle). De Casteljau's algorithm gives an alternative way to compute such Bézier surfaces. In suggestive matrix notation, it takes the following form

$$\begin{aligned} \mathfrak{b}_{ij}^0 &= b_{ij}, & i, j = 0, \dots, K, \\ \mathfrak{b}_{ij}^k &= (1 - t_1 \quad t_1) \begin{pmatrix} \mathfrak{b}_{i,j}^{k-1} & \mathfrak{b}_{i,j+1}^{k-1} \\ \mathfrak{b}_{i+1,j}^{k-1} & \mathfrak{b}_{i+1,j+1}^{k-1} \end{pmatrix} \begin{pmatrix} 1 - t_2 \\ t_2 \end{pmatrix}, & k = 1, \dots, K,\ i, j = 0, \dots, K - k, \end{aligned} \tag{7}$$

and yields $\mathfrak{b}_{00}^K = \beta_K(t_1, t_2; (b_{ij})_{i,j=0,\dots,K})$. Here again, points $\mathfrak{b}_{ij}^k$ are obtained as convex combinations of $\mathfrak{b}_{ij}^{k-1}$, $\mathfrak{b}_{ij+1}^{k-1}$, $\mathfrak{b}_{i+1j}^{k-1}$ and $\mathfrak{b}_{i+1j+1}^{k-1}$. This combination leads once more to a simple geometric interpretation (Figure 3 right).

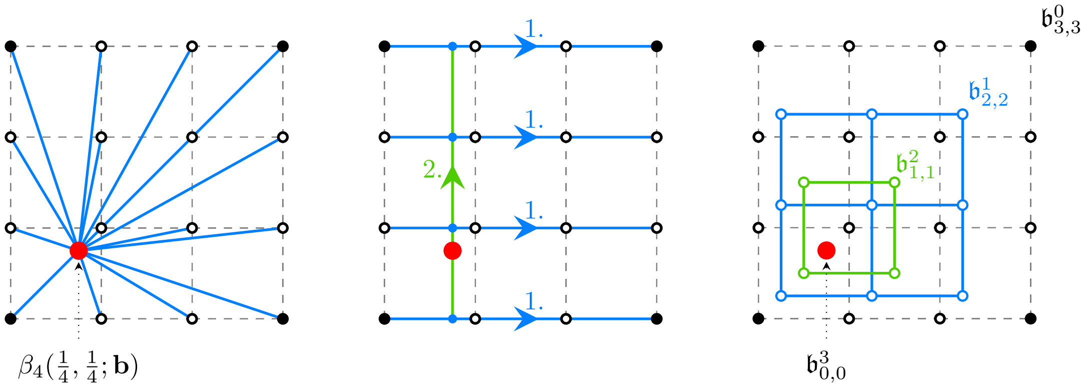


Figure 3: Computation of a cubic Bézier surface via (5) as a weighted mean of the control points (left), via (6) through a one-parameter family of Bézier curves (middle), and via de Casteljau's algorithm (7) (right).

Two Bézier surfaces $\beta_K(t_1, t_2; (b_{ij}^{\mathrm{l}})_{i,j=0,\dots,K})$ and $\beta_K(t_1, t_2; (b_{ij}^{\mathrm{r}})_{i,j=0,\dots,K})$ can be joined $\mathcal{C}^k$-continuously in t_1-direction via

$$\mathfrak{B} : [0,2] \times [0,1] \to \mathbb{R}^n : \quad (t_1, t_2) \mapsto \begin{cases} \beta_K(t_1, t_2; (b_{ij}^{\mathrm{l}})_{i,j=0,\dots,K}) & \text{if } t_1 \in [0,1], \\ \beta_K(t_1 - 1, t_2; (b_{ij}^{\mathrm{r}})_{i,j=0,\dots,K}) & \text{if } t_1 \in (1,2], \end{cases} \tag{8}$$

if for all $j = 1, \dots, K$ the sequence pairs $(b_{j,0}^{\mathrm{l}}, \dots, b_{j,K}^{\mathrm{l}}), (b_{j,0}^{\mathrm{r}}, \dots, b_{j,K}^{\mathrm{r}})$ induce a one-dimensional $\mathcal{C}^k$-continuous Bézier spline via (4). An analogous condition holds if the two surfaces are to be matched smoothly in t_2-direction (see [8, § 16.1]).

3 Piecewise-Bézier surfaces on manifolds

The generalization of Bézier curves to a manifold setting was previously studied. For instance, Lin and Walker [17] proposed a manifold version of de Casteljau's algorithm. The conditions of continuity and derivability were studied by Popiel and Noakes [20]. Here, we aim to generalize the concept of a Bézier surface to a Riemannian manifold $\mathcal{M}$. To this end, we briefly introduce some standard notions of Riemannian geometry (a more detailed exposition is found in standard textbooks, *e.g.*, [7]).

The Riemannian metric $g_y(\cdot,\cdot)$ in a point $y \in \mathcal{M}$ is an inner product on the tangent space $T_y\mathcal{M}$ to $\mathcal{M}$ in y, and it depends smoothly on y. The length and energy of a path $\gamma : [0,1] \to \mathcal{M}$ are defined as

$$L[\gamma] = \int_0^1 \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}\, \mathrm{d}t\,, \tag{9}$$

$$E[\gamma] = \int_0^1 g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))\, \mathrm{d}t\,, \tag{10}$$

where the dot denotes (time) differentiation.

We call *geodesics* the paths γ that minimize the energy $E[\gamma]$ for fixed end points $\gamma(0) = y_0 \in \mathcal{M}$, $\gamma(1) = y_1 \in \mathcal{M}$. The length $L[\gamma]$ is also minimized by those paths [7].

The Riemannian distance $\mathrm{d}(y_0, y_1)$ between any two points y_0, y_1 is the minimum path length of a path from y_0 to y_1. For $\mathrm{d}(y_0, y_1)$ small enough, the connecting geodesic is unique. We shall assume that $\mathcal{M}$ is finite-dimensional, connected, and geodesically complete. By the Hopf–Rinow theorem this implies that every two points on $\mathcal{M}$ can be connected by a geodesic. The logarithm of y_1 with respect to y_0 is defined as the initial velocity of the geodesic γ from y_0 to y_1,

$$\log_{y_0}(y_1) = \dot{\gamma}(0). \tag{11}$$

The exponential map of $v \in T_{y_0}\mathcal{M}$ is the point reached at $t = 1$ by the geodesic γ starting from y_0 with initial velocity v,

$$\exp_{y_0}(v) = \gamma(1). \tag{12}$$

Bézier surfaces can be transfered to a manifold setting in different ways, where each approach generalizes a particular evaluation scheme for Bézier surfaces in Euclidean space. In this section, we introduce three possible definitions of Bézier surfaces in a Riemannian manifold $\mathcal{M}$ (in fact, the definitions even extend to arbitrary metric spaces). We also examine the conditions and techniques to achieve $\mathcal{C}^0$ and $\mathcal{C}^1$-continuity when patching multiple Bézier surfaces together. Each approach has its own advantages and disadvantages.

3.1 Bézier surface definitions based on geodesic averaging

To extend Bézier surfaces to metric spaces $\mathcal{M}$, we mainly use the weighted average between points $y_1, \dots, y_n \in \mathcal{M}$ as core concept. This tool can replace naturally not only convex combinations, but also multilinear interpolations. Thanks to this, we will propose three ways to compute Bézier surfaces on manifolds and analyze their well-posedness.

Definition 1 (Weighted geodesic average). *Let $n \in \mathbb{N}$. A* weighted geodesic average *of $y_1, \dots, y_n \in \mathcal{M}$ for convex combination weights $w_1, \dots, w_n \in [0,1]$ with $\sum_{i=1}^n w_i = 1$ is any point $y \in \mathcal{M}$ solving*

$$\min_{y \in \mathcal{M}} \sum_{i=1}^n w_i \mathrm{d}^2(y_i, y)\,.$$

If the minimizer exists and is unique, we denote it by $\mathrm{av}[(y_1, \dots, y_n), (w_1, \dots, w_n)]$.

Note that weighted averages in metric spaces can be traced back to Fréchet [10]. On Riemannian manifolds, they have been analyzed under various names—Riemannian center of mass, Riemannian barycenter, Karcher mean, Riemannian average, geodesic average—in particular by Karcher (see, *e.g.*, [16]).

Remark 2 (Convex combinations and multilinear interpolation). In Euclidean space, the weighted geodesic average is the same as a convex combination of the points $y_1, \dots, y_n \in \mathbb{R}^d$. Indeed, it is straightforward to check that

$$\operatorname{av}[(y_1, \dots, y_n), (w_1, \dots, w_n)] = \sum_{i=1}^{n} w_i y_i \,.$$

Likewise, the bilinear interpolation of points $y_{ij} \in \mathbb{R}^d$, $i, j \in \{0, 1\}$, at coordinates $(t_1, t_2) \in [0, 1]^2$ can be expressed as a weighted geodesic average

$$\begin{aligned}(1-t_1)(1-t_2)y_{00} &+ (1-t_1)t_2y_{01} + t_1(1-t_2)y_{10} + t_1t_2y_{11} \\ &= \operatorname{av}[(y_{00}, y_{01}, y_{10}, y_{11}), ((1-t_1)(1-t_2), (1-t_1)t_2, t_1(1-t_2), t_1t_2)] \,.\end{aligned}$$

◁

Remark 3 (Weighted geodesic average of two points). For two points $y_1, y_2 \in \mathcal{M}$ with weights $1-w, w \in [0, 1]$ and any $y \in \mathcal{M}$, the Young-type inequality $(1-w)^2\mathrm{d}^2(y_1, y) + w^2\mathrm{d}^2(y_2, y) \geq 2(1-w)w\mathrm{d}(y_1, y)\mathrm{d}(y_2, y)$ implies

$$\begin{aligned}(1-w)\mathrm{d}^2(y_1, y) + w\mathrm{d}^2(y_2, y) &\geq (1-w)w(\mathrm{d}(y_1, y) + \mathrm{d}(y_2, y))^2 \\ &\geq (1-w)w\mathrm{d}^2(y_1, y_2) \\ &= (1-w)\mathrm{d}^2(y_1, \hat{y}) + w\mathrm{d}^2(y_2, \hat{y})\end{aligned}$$

for $\hat{y}$ the point on the connecting geodesic between y_1 and y_2 with $\mathrm{d}(y_1, \hat{y}) = w\mathrm{d}(y_1, y_2)$ and $\mathrm{d}(y_2, \hat{y}) = (1-w)\mathrm{d}(y_1, y_2)$. Thus we obtain the explicit formula

$$\operatorname{av}[(y_1, y_2), (1-w, w)] = \hat{y} = \exp_{y_1}\left(w\log_{y_1}(y_2)\right) \,. \tag{13}$$

◁

The definitions of Bézier surfaces on manifolds are generalizations of (5)–(7). Note that equation (5) represents a convex combination and each iteration step in equation (7) represents a bilinear interpolation.

Definition 4 (Generalized Bézier surface). *Given control points $b_{ij} \in \mathcal{M}$, $i, j = 0, \dots, K$, we define a corresponding* generalized Bézier surface *of type I–III via*

$$\beta_K^{I}(t_1, t_2; (b_{ij})_{i,j=0,\dots,K}) = \operatorname{av}[(b_{ij})_{i,j=0,\dots,K}, (B_{iK}(t_1)B_{jK}(t_2))_{i,j=0,\dots,K}] \,, \tag{14}$$

$$\beta_K^{II}(t_1, t_2; (b_{ij})_{i,j=0,\dots,K}) = \beta_K(t_1; (\beta_K(t_2; (b_{ij})_{j=0,\dots,K}))_{i=0,\dots,K}) \,, \tag{15}$$

$$\beta_K^{III}(t_1, t_2; (b_{ij})_{i,j=0,\dots,K}) = \mathfrak{b}_{00}^K \,, \tag{16}$$

where $\beta_K(\cdot\,; (b_m)_{m=0,\dots,K})$ denotes a Bézier curve in $\mathcal{M}$ and where $\mathfrak{b}_{00}^K$ is defined recursively via de Casteljau's algorithm,

$$\begin{aligned}\mathfrak{b}_{ij}^0 &= b_{ij}, & i, j = 0, \dots, K, \\ \mathfrak{b}_{ij}^k &= \operatorname{av}[(\mathfrak{b}_{ij}^{k-1}, \mathfrak{b}_{i,j+1}^{k-1}, \mathfrak{b}_{i+1,j}^{k-1}, \mathfrak{b}_{i+1,j+1}^{k-1}), (w_{00}, w_{01}, w_{10}, w_{11})] \,, & k = 1, \dots, K,\ i, j = 0, \dots, K-k,\end{aligned}$$

with weights

$$w_{00} = (1-t_1)(1-t_2),\ w_{01} = (1-t_1)t_2,\ w_{10} = t_1(1-t_2),\ w_{11} = t_1t_2.$$

All these three types reduce to the classical Bézier surface if the manifold is the Euclidean space $\mathcal{M} = \mathbb{R}^d$. However, they will generally differ from each other on general manifolds (Figure 4).

In the above definition of β_K^{II} we have not yet specified how Bézier curves in $\mathcal{M}$ are defined. Here we can again either employ a generalization of (1),

$$\beta_K(t;(b_j)_{j=0,\dots,K}) = \mathrm{av}[(b_j)_{j=0,\dots,K},(B_{jK}(t))_{j=0,\dots,K}]\,,$$

or use a definition via de Casteljau's algorithm as in [17, 11], where $\beta_K(t;(b_j)_{j=0,\dots,K}) = \mathfrak{b}_0^K$ for

$$\begin{aligned} \mathfrak{b}_j^0 &= b_j, & j &= 0,\dots,K,\\ \mathfrak{b}_j^k &= \mathrm{av}[(\mathfrak{b}_j^{k-1},\mathfrak{b}_{j+1}^{k-1}),(1-t,t)], & k &= 1,\dots,K,\ j=0,\dots,K-k. \end{aligned}$$

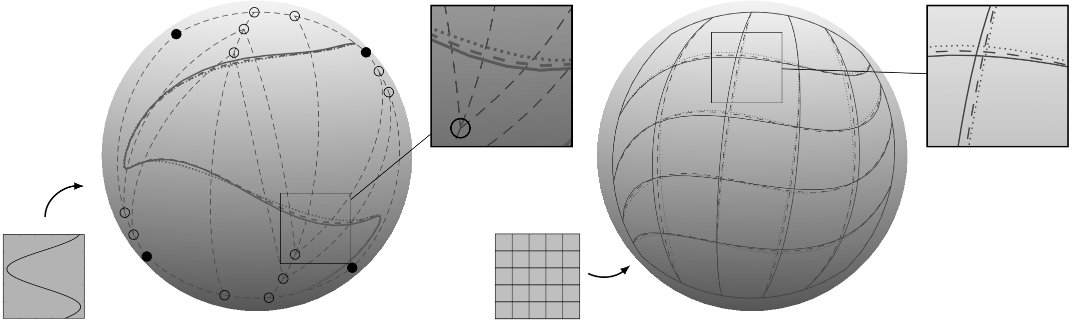


Figure 4: The Bézier surfaces of type *I* to *III* differ from each other. Here this is visualized on the sphere $\mathbb{S}^2$ by mapping a curve (left) or a grid (right) on the parameterization domain onto the Bézier surface. Type *I* to *III* correspond to the solid, dashed, and dotted lines, respectively. Note that the Bézier surface boundaries coincide, though. The control and interpolation points of the Bézier surface are displayed in the left picture; the gray dashed lines indicate the control point grid.

All three approaches have their advantages and disadvantages.

The evaluation of β_K^I entails solving a rather complex optimization problem. However, it does not suffer from the drawbacks of the other two approaches.

In comparison to β_K^I, an evaluation of β_K^{III} via de Casteljau's algorithm only requires the comparatively simple computation of weighted geodesic averages with four points. Unfortunately there does not seem to be a straightforward way to patch multiple Bézier surfaces of type *III* together $\mathcal{C}^1$-continuously (this will be discussed in Section 3.4).

Finally, the evaluation of β_K^{II} requires only weighted averages of two points as it is based on the one-dimensional de Casteljau's algorithm. When simple analytical formulas exist for the Riemannian exponential and logarithm (*e.g.*, for $\mathcal{M}=\mathbb{S}^m$, see also [22]), this method is very avantageous since the averaging can be based on (13). However, unlike the surfaces of type *I* and *III*, the definition of β_K^{II} is not symmetric, because it does not satisfy the relation

$$\beta_K(t_1,t_2;(b_{ij})_{i,j=0,\dots,K}) = \beta_K(t_2,t_1;(b_{ij})_{i,j=0,\dots,K}^T)\,.$$

In the definition of β_K^{II}, instead of first computing the Bézier curve with respect to t_2, one could alternatively compute first the Bézier curves $\beta_K(t_1;(b_{ij})_{i=0,\dots,K})$ for each j to obtain new control points for a Bézier curve which is then evaluated at t_2,

$$\beta_K(t_2;(\beta_K(t_1;(b_{ij})_{i=0,\dots,K}))_{j=0,\dots,K})\,. \tag{17}$$

In general, both approaches will yield different surfaces (Figure 5) so that there is a choice to be made about along which variable of (t_1,t_2) to interpolate first.

3.2 Well-posedness and smoothness of Bézier surfaces

In order to analyze the well-posedness of β_K^I, β_K^{II}, and β_K^{III} (Definition 4), we now present generalizations of standard notions in Riemannian geometry.

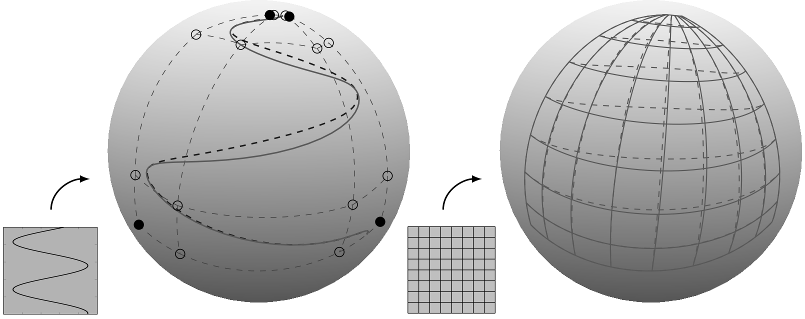


Figure 5: The two possible construction methods (15) and (17) for Bézier surfaces of type *II* produce in general different results (their boundaries always coincide, however). Here the difference is shown on the sphere with the same conventions as in Figure 4: the solid line corresponds to computing vertical splines first, the dashed line to computing horizontal splines first.

Definition 5 (Multigeodesic convexity)**.** *A subset $U \subset \mathcal{M}$ is called* multigeodesically convex *if it contains any weighted geodesic average of any of its points. The* multigeodesically convex hull*, $\operatorname{co}(U)$, of a set $U \subset \mathcal{M}$ is the smallest multigeodesically convex set $C \subset \mathcal{M}$ containing U.*

Remark 6. A multigeodesically convex set is a generalization of a geodesically convex set, *i.e.*, a set in which only weighted geodesic averages of pairs of points are considered. Thus, by definition, any multigeodesically convex set is also geodesically convex. However, the opposite inclusion is false in general: consider for instance $\mathbb{R}^3$ with the Euclidean metric $g_x(v,v) = \|v\|^2$, only with the difference that the metric is smaller, say $g_x(v,v) = \alpha\|v\|^2$ with some $0 < \alpha < 1$, for x in a neighborhood of the three lines connecting the origin with $y_i = (\cos\frac{2i\pi}{3}, \sin\frac{2i\pi}{3}, 1)$, $i = 1,2,3$ (Figure 6). We claim that for $\alpha \in (\sqrt{3/2}-1, \sqrt{2}-1)$ the triangle spanned by $2y_1$, $2y_2$, and $2y_3$ is geodesically convex, but not multigeodesically convex. Indeed, it is straightforward to check that for $\alpha < \sqrt{2}-1$ the average of $2y_1$, $2y_2$, and $2y_3$ is close to the origin,

$$\operatorname{av}[(2y_1, 2y_2, 2y_3), (\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3})] \approx (0,0,0)\,.$$

Since the triangle does not contain this average, it cannot be multigeodesically convex. However, the shortest geodesic between any two points of the triangle is fully contained in that triangle, so it is geodesically convex. In particular the geodesic between two triangle corners, which is the most critical case, is the straight connecting line as long as $\alpha > \sqrt{3/2}-1$. ◁

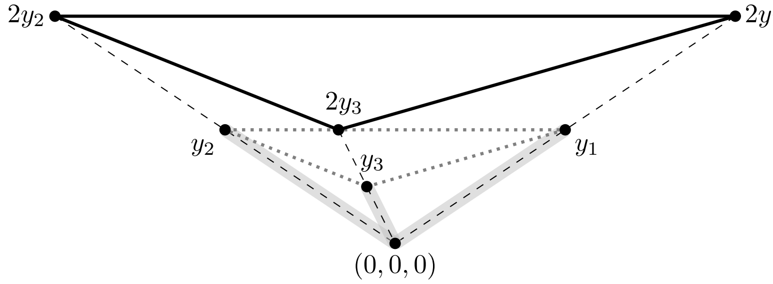


Figure 6: Illustration of the counterexample given in Remark 6. The gray color indicates areas where the metric is smaller than the Euclidean metric.

Remark 7. In our definition of the convex hull, we require *every* weighted average to be contained (and not just one of the potentially multiple possibilities). This is different from, *e.g.*, [15, Def. 3.3.1]. Thus, the multigeodesically convex hull $\mathrm{co}(U)$ always exists as the intersection of all multigeodesically convex sets containing U. ◁

Remark 8. The multigeodesically convex hull $\mathrm{co}(U)$ can in general not be obtained as the set of all weighted geodesic averages of points in U: taking $U = \{y_1, y_2, y_3\}$, the latter will contain the origin, but not the point $(0, 0, 1)$, while the former contains both. This differs from Euclidean space. ◁

The following theorem ensures that one can always find a weighted geodesic average, thereby proving part of the well-posedness of geodesic averages (which is necessary to justify our approach). The remaining issues of uniqueness and smooth data dependence can only be expected to hold under more restrictive conditions and are treated further below.

Theorem 9 (Existence of weighted geodesic averages)**.** *Let $\mathcal{M}$ be a smooth finite-dimensional geodesically complete Riemannian manifold. For any points $y_1, \ldots, y_n \in \mathcal{M}$ and weights $w_1, \ldots, w_n \in [0, 1]$, $\sum_{i=1}^n w_i = 1$, a weighted geodesic average exists.*

Proof. This is a simple application of the direct method in the calculus of variations. Set $J[y] = \sum_{i=1}^n w_i \mathrm{d}^2(y_i, y)$ and consider a minimizing sequence y^j, $j \in \mathbb{N}$, with $J[y^j] \to \inf_{y\in\mathcal{M}} J[y]$ monotonically as $j \to \infty$. Without loss of generality, let $w_1 \neq 0$. Due to $\mathrm{d}(y_1, y^j) \leq \frac{1}{w_1} J[y^j] \leq \frac{1}{w_1} J[y^1]$, the sequence y^j is bounded. Hence the Hopf–Rinow theorem implies that y^j is precompact, *i.e.*, y^j converges (up to a subsequence, which we again denote by y^j for simplicity) to some $y \in \mathcal{M}$. The limit y is a weighted geodesic average, since by the continuity of $\mathrm{d}(\cdot, \cdot)$ we have $J[y] = \lim_{j\to\infty} J[y^j] = \inf_{y\in\mathcal{M}} J[y]$. □

Remark 10. An analogous result for infinite-dimensional manifolds $\mathcal{M}$ is much harder to obtain and probably requires assumptions similar to the ones in [23] for the existence of geodesics. There the manifold needs to be modeled over a Hilbert space V which is compactly embedded in another Banach space Y. The inner product of V needs to be globally equivalent to the Riemannian metric $g_y(\cdot, \cdot)$, and $g_y(\cdot, \cdot)$ has to depend Lipschitz-continuously on $y \in Y$. ◁

Definition 11 (Proper subset)**.** *We call a subset $U \subset \mathcal{M}$* proper *if the weighted geodesic averages between any finitely many points in U are unique and smoothly depend on the points and the weights.*

Proposition 12 (Existence of proper neighborhoods)**.** *For every $y \in \mathcal{M}$ there is a proper multigeodesically convex neighborhood.*

Proof. Let us again set $J[y] = \sum_{i=1}^m w_i \mathrm{d}^2(y_i, y)$. For all $\rho > 0$ small enough the ball $B_\rho(y)$ of radius ρ around y is geodesically convex. By [16, Thm. 1.2], for $y_1, \ldots, y_n \in B_\rho(y)$ the energy J has a (local) minimizer in the interior of $B_\rho(y)$. Now let Δ denote an upper bound on the sectional curvatures of $\mathcal{M}$ in a neighborhood of y and let r be the injectivity radius of $\exp_y$. If $\rho \leq \rho_0$ with $\rho_0 = \min(r, \frac{\pi}{4\sqrt{\Delta}})$ for $\Delta > 0$ and $\rho_0 = r$ else, J is strictly convex on $B_\rho(y)$ by [16, Thm. 1.2] and thus the minimizer unique.

Now take $U = B_{\rho_0/3}(y)$, then for any points $y_1, \ldots, y_n \in U$ and any convex combination weights $w_1, \ldots, w_n$, the energy J has a unique global minimizer which lies inside U. Indeed, the minimizer cannot lie outside $B_{\rho_0}(y)$ since for any $\hat{y} \in \mathcal{M} \setminus B_{\rho_0}(y)$ we have $J[\hat{y}] \geq n(\frac{2}{3}\rho_0)^2 > J[y_1]$, and inside $B_{\rho_0}(y)$ the energy J is strictly convex. Thus U is multigeodesically convex and every weighted average of points from U is unique.

As for the smoothness of averages in U, note that by [16, Thm. 1.2] the weighted geodesic average $b = \mathrm{av}[(y_1, \ldots, y_n), (w_1, \ldots, w_n)]$ is characterized as that point $\tilde{b} \in \mathcal{M}$ for which $0 = \mathrm{grad} J[\tilde{b}] = -2 \sum_{k=1}^n w_k \log_{\tilde{b}} y_k$. For an easier exposition we replace $\tilde{b} = \exp_b v$ for some $v \in T_b\mathcal{M}$. Then the above optimality condition turns into

$$0 = \sum_{k=1}^n w_k \log_{\exp_b v} y_k =: F(v, y_1, \ldots, y_n, w_1, \ldots, w_n)\,.$$

By the implicit function theorem, this condition can be solved for v in a neighborhood of $(0, y_1, \dots, y_n, w_1, \dots, w_n)$, if $\frac{\partial F}{\partial v}(0, y_1, \dots, y_n, w_1, \dots, w_n)$ is regular, and the resulting function $v(y_1, \dots, y_n, w_1, \dots, w_n)$ is smooth if F is smooth. Since b is unique, in a neighborhood of b, the function $\exp_b v$ describes the weighted geodesic average for different input points and weights.

The smoothness of F is clear, based on the smoothness of Riemannian exponential and logarithm. The regularity of $\frac{\partial F}{\partial v}(0, y_1, \dots, y_n, w_1, \dots, w_n)$ follows directly from [16, Thm. 1.2], which also states that $\frac{\mathrm{d}^2}{\mathrm{d}t^2} J[y(t)] \geq c\|\dot{y}\|^2$ for any geodesic y, where the constant $c > 0$ depends on the sectional curvature of the manifold and on ρ_0. Thus we have

$$g_b\left(\frac{\partial F}{\partial v}(0, y_1, \dots, y_n, w_1, \dots, w_n)v, v\right) = \frac{\mathrm{d}^2}{\mathrm{d}t^2} J[\exp_b(tv)]_{t=0} \geq c\|v\|^2$$

so that $\frac{\partial F}{\partial v}(0, y_1, \dots, y_n, w_1, \dots, w_n)$ is symmetric positive definite and thus invertible by the Lax–Milgram theorem. □

Corollary 13 (Existence of Bézier surfaces). *Let control points $(b_{ij})_{i,j=0,\dots,K} \in \mathcal{M}$ be given. If $\operatorname{co}(\{b_{ij}\}_{i,j=0,\dots,K})$ is proper, then β_K^{I}, β_K^{II}, and β_K^{III} in Definition 4 are well-defined and define smooth surfaces in $\operatorname{co}(\{b_{ij}\}_{i,j=0,\dots,K}) \subset \mathcal{M}$.*

Proof. First we note that, if the weighted geodesic averages involved in the computation of the Bézier surfaces are well-defined (*i.e.*, they exist and are unique), then the Bézier surfaces are well-defined and lie in the convex hull of their control points by definition. The existence of all averages follows from Theorem 9, their uniqueness and smoothness from the properness of $\operatorname{co}(\{b_{ij}\}_{i,j=0,\dots,K})$. □

3.3 $\mathcal{C}^0$-patching

As in Euclidean space, two generalized Bézier surfaces can be patched together via the generalization of (8) to a manifold $\mathcal{M}$,

$$\mathfrak{B}^Y : [0,2] \times [0,1] \to \mathcal{M} : \quad (t_1, t_2) \mapsto \begin{cases} \beta_K^Y(t_1, t_2; (b_{ij}^{\mathrm{l}})_{i,j=0,\dots,K}) & \text{if } t_1 \in [0,1], \\ \beta_K^Y(t_1 - 1, t_2; (b_{ij}^{\mathrm{r}})_{i,j=0,\dots,K}) & \text{if } t_1 \in (1,2], \end{cases} \tag{18}$$

for $Y = I, II$, or III (and analogously for patching in the t_2-direction). The Bézier spline will be $\mathcal{C}^0$-continuous under the same conditions as in the Euclidean space.

Theorem 14 ($\mathcal{C}^0$-continuity). *For $Y = I, II$, and III, the patched Bézier surface $\mathfrak{B}^Y$ in $\mathcal{M}$ is continuous if $b_{K,j}^{\mathrm{l}} = b_{0,j}^{\mathrm{r}}$ for $j = 0, \dots, K$.*

Proof. The generalized Bézier surfaces (Definition 4) are smooth (Corollary 13), hence it remains to consider the continuity at the interface of the two domains, *i.e.*, at $t_1 = 1$. From the definition of the Bézier surfaces β^Y, it immediately follows that only the control points $b_{K,j}^{\mathrm{l}}$ and $b_{0,j}^{\mathrm{r}}$ are involved in the corresponding weighted averages. Furthermore, $\beta_K^Y(1, t_2; (b_{ij}^{\mathrm{l}})_{i,j=0,\dots,K})$ and $\beta_K^Y(0, t_2; (b_{ij}^{\mathrm{r}})_{i,j=0,\dots,K})$ are the Bézier curves in $\mathcal{M}$ defined by the control points $(b_{K,j}^{\mathrm{l}})_{j=0,\dots,K}$ and $(b_{0,j}^{\mathrm{r}})_{j=0,\dots,K}$, respectively. Thus, if those control points coincide, the corresponding curves coincide too. □

Remark 15. In Riemannian spaces where logarithm and exponential maps are very easy to compute, one might be tempted to replace the weighted geodesic averages in the definitions of β_K^{I} and β_K^{III} by the similar and simpler expression

$$\operatorname{av}[(y_1, \dots, y_n), (w_1, \dots, w_n)] \approx \exp_y\left(w_1 \log_y y_1 + \ldots + w_n \log_y y_n\right),$$

which performs a weighted average in the tangent space at some $y \in \mathcal{M}$. For instance, in the definition of β_K^{III} we might redefine $\mathfrak{b}_{ij}^k = \exp_{\mathfrak{b}_{ij}^{k-1}}\left(\sum_{(r,s)\in\{i,i+1\}\times\{j,j+1\}} \log_{\mathfrak{b}_{ij}^{k-1}} w_{rs}\mathfrak{b}_{rs}^{k-1}\right)$, in which case the following can be observed.

- The curve $\beta_K^{III}(0, t_2; (b_{ij}^{\mathrm{r}})_{i,j=0,\ldots,K})$ is still a Bézier curve with control points $(b_{0,j}^{\mathrm{r}})_{j=0,\ldots,K}$. The involved weighted geodesic averaging reduces to two-point averages at each step of the recursive process. By (13), the two-point averages computed by the new or the old formula are identical.
- The curve $\beta_K^{III}(1, t_2; (b_{ij}^{\mathrm{l}})_{i,j=0,\ldots,K})$ is not a Bézier curve with control points $(b_{K,j}^{\mathrm{r}})_{j=0,\ldots,K}$. Here, the averaging is based on the tangent space at a different point than the two of which the weighted average is computed.

As a consequence of this second observation, $\mathcal{C}^0$-continuous patching of Bézier surfaces is no longer possible (Figure 7), even in symmetric spaces such as the sphere. ◁

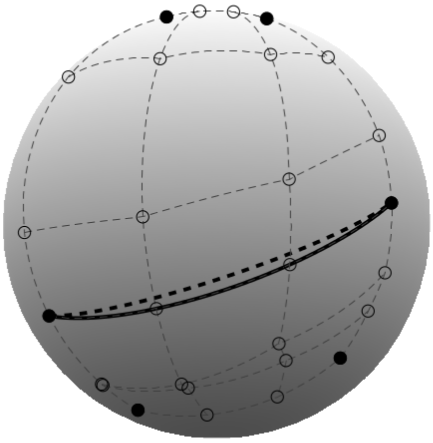


Figure 7: Two cubic Bézier surfaces computed on the sphere with de Casteljau's algorithm modified as in Remark 15. At the interface where control points coincide, the surfaces do not match up continuously due to the simplification of weighted geodesic averaging.

3.4 $\mathcal{C}^1$-patching challenges

In contrast to C^0-continuity, which is easily achieved, C^1-patching represents a challenge. Indeed, the corresponding conditions in the Euclidean space [8] cannot be used as we will see. In the following we will always choose the outmost control points of each Bézier surface patch such that $\mathcal{C}^0$-continuity is ensured. The Euclidean conditions $b_{K,j}^{\mathrm{l}} = \frac{b_{K-1,j}^{\mathrm{l}} + b_{1,j}^{\mathrm{r}}}{2}$ for $\mathcal{C}^1$-continuity now generalize to the Riemannian setting as

$$b_{K,j}^{\mathrm{l}} = \mathrm{av}[(b_{K-1,j}^{\mathrm{l}}, b_{1,j}^{\mathrm{r}}), (\tfrac{1}{2}, \tfrac{1}{2})] \tag{19}$$

for all j. However, there are multiple problems with this generalization. Consider four Bézier surfaces that are patched together as a bidimensional spline $\mathfrak{B} : [0,2] \times [0,2] \to \mathcal{M}$ via

$$(t_1, t_2) \mapsto \begin{cases} \beta_K^Y(t_1, t_2; (b_{ij}^{\mathrm{bl}})_{i,j=0,\ldots,K}) & \text{if } (t_1,t_2) \in [0,1] \times [0,1], \\ \beta_K^Y(t_1 - 1, t_2; (b_{ij}^{\mathrm{br}})_{i,j=0,\ldots,K}) & \text{if } (t_1,t_2) \in [1,2] \times [0,1], \\ \beta_K^Y(t_1, t_2 - 1; (b_{ij}^{\mathrm{tl}})_{i,j=0,\ldots,K}) & \text{if } (t_1,t_2) \in [0,1] \times [1,2], \\ \beta_K^Y(t_1 - 1, t_2 - 1; (b_{ij}^{\mathrm{tr}})_{i,j=0,\ldots,K}) & \text{if } (t_1,t_2) \in [1,2] \times [1,2], \end{cases} \tag{20}$$

where the superscripts $\mathrm{bl}, \mathrm{br}, \mathrm{tl}, \mathrm{tr}$ stand for *bottom left*, *bottom right*, *top left*, and *top right* respectively. The conditions of type (19) for $\mathcal{C}^1$-continuity near the midpoint now read (*cf.*

Figure 8 left)

$$\begin{aligned}
b^{\mathrm{bl}}_{K,K-1} &= \mathrm{av}[(b^{\mathrm{bl}}_{K-1,K-1}, b^{\mathrm{br}}_{1,K-1}), (\tfrac{1}{2}, \tfrac{1}{2})],\\
b^{\mathrm{bl}}_{K-1,K} &= \mathrm{av}[(b^{\mathrm{bl}}_{K-1,K-1}, b^{\mathrm{tl}}_{K-1,1}), (\tfrac{1}{2}, \tfrac{1}{2})],\\
b^{\mathrm{bl}}_{K,K} &= \mathrm{av}[(b^{\mathrm{bl}}_{K-1,K}, b^{\mathrm{br}}_{1,K}), (\tfrac{1}{2}, \tfrac{1}{2})],\\
b^{\mathrm{bl}}_{K,K} &= \mathrm{av}[(b^{\mathrm{bl}}_{K,K-1}, b^{\mathrm{tl}}_{K,1}), (\tfrac{1}{2}, \tfrac{1}{2})],\\
b^{\mathrm{tl}}_{K,1} &= \mathrm{av}[(b^{\mathrm{tl}}_{K-1,1}, b^{\mathrm{tr}}_{1,1}), (\tfrac{1}{2}, \tfrac{1}{2})],\\
b^{\mathrm{br}}_{1,K} &= \mathrm{av}[(b^{\mathrm{br}}_{1,K-1}, b^{\mathrm{tr}}_{1,1}), (\tfrac{1}{2}, \tfrac{1}{2})].
\end{aligned} \tag{21}$$

These are six equations in nine control points. In Euclidean space the equations are linearly dependent so that there are only five independent equations and one can choose four of the nine control points as independent variables. In the Riemannian setting, however, the linear dependence of the six equations turns into an incompatibility: in general it is not possible to satisfy all six equations (unless several control points collapse to single points in $\mathcal{M}$).

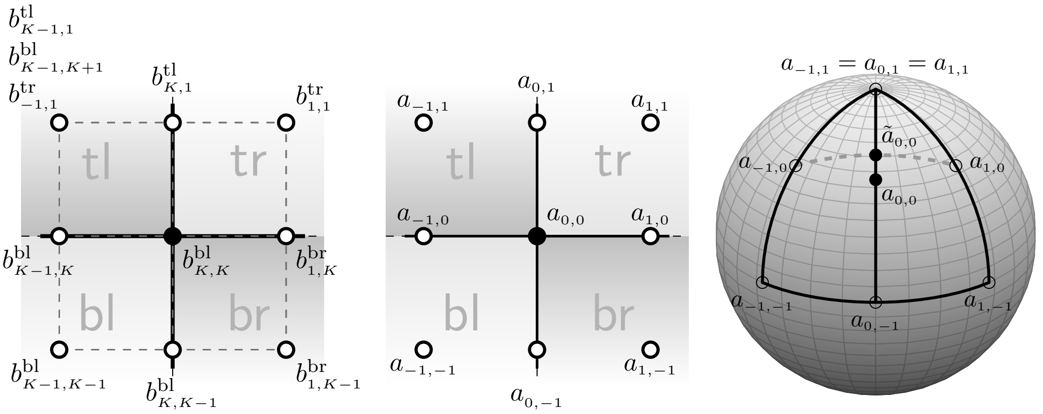


Figure 8: Left: $\mathcal{C}^1$-conditions of equation (21) at the interface of four Bézier surface patches. Along each dashed line, the middle point should be the average of the end points. Middle: Renaming scheme for the control points. Right: Illustration of the counterexample from Proposition 16.

A special situation occurs in symmetric spaces (spaces $\mathcal{M}$ such that for every $\hat{y} \in \mathcal{M}$ there exists an isometry $I_{\hat{y}} : \mathcal{M} \to \mathcal{M}$ which maps any geodesic $y : [0,1] \to \mathcal{M}$ with $y(\frac{1}{2}) = \hat{y}$ onto $I_{\hat{y}}(y) : t \mapsto y(1-t)$). As the following proposition shows, in those spaces one can always find a nondegenerate set of control points satisfying the above six equations. However, instead of four one can only choose three of the nine control points freely. In the following propositions, for simplicity we will write

$$b^{\mathrm{br}}_{i,K+j} = b^{\mathrm{tl}}_{K+i,j} = b^{\mathrm{tr}}_{ij} = b^{\mathrm{bl}}_{K+i,K+j} = a_{ij},$$

for $i, j \in \{-1, 0, 1\}$ (see Figure 8, middle). The control points with indices below 0 or beyond K simply indicate control points in the neighboring Bézier surface patches; a_{ij} refers to control points in a specific area where four patches meet.

Proposition 16. *Let $\mathcal{M}$ be a symmetric space. Of the nine control points a_{ij}, $i, j = \{-1, 0, 1\}$, let $a_{0,0}, a_{i_1,j_1}, a_{i_2,j_2}$ be given with $(i_1, j_1) \neq -(i_2, j_2)$, $|i_1| = |j_1|$. One can find the other six control points such that (21) is satisfied.*

If instead four control points are given initially, those can in general not be complemented with five other control points such that (21) is satisfied.

Proof. To prove the first part, we complement the six equations (21) with the following additional two equations,

$$a_{0,0} = \mathrm{av}[(a_{-1,-1}, a_{1,1}), (\tfrac{1}{2}, \tfrac{1}{2})], \quad a_{0,0} = \mathrm{av}[(a_{-1,1}, a_{1,-1}), (\tfrac{1}{2}, \tfrac{1}{2})].$$

Now we have eight equations for the nine control points, where each equation expresses one control point as the midpoint on the geodesic between two other control points. The equations of the form $a_{0,0} = \mathrm{av}[(a_{ij}, a_{-i,-j}), (\frac{1}{2}, \frac{1}{2})]$ just establish the relation

$$a_{ij} = I_{a_{0,0}}(a_{-i,-j}) \qquad i,j \in \{-1,0,1\}$$

for the isometry $I_{a_{0,0}}$ of the symmetric space. Thus, there are actually only two unknown control points left, the others being fixed by the above identity. Furthermore, the remaining four equations,

$$\begin{aligned} & a_{-1,0} = \mathrm{av}[(a_{-1,-1}, a_{-1,1}), (\tfrac{1}{2}, \tfrac{1}{2})] \qquad \text{and} \quad a_{1,0} = \mathrm{av}[(a_{1,-1}, a_{1,1}), (\tfrac{1}{2}, \tfrac{1}{2})] \\ \text{as well as} \quad & a_{0,-1} = \mathrm{av}[(a_{-1,-1}, a_{1,-1}), (\tfrac{1}{2}, \tfrac{1}{2})] \qquad \text{and} \quad a_{0,1} = \mathrm{av}[(a_{-1,1}, a_{1,1}), (\tfrac{1}{2}, \tfrac{1}{2})] \end{aligned}$$

are redundant due to the identity stated before. It is therefore straightforward to check that two of them may be chosen to determine the desired two control points.

For the second part, a simple counterexample suffices. Consider the unit sphere $\mathcal{M} = \mathbb{S}^2$ with $a_{-1,1} = a_{1,1} = (0,0,1)^T$, $a_{0,0} = (\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2})^T$, and $a_{-1,0} = (\frac{1}{2}, -\frac{1}{2}, \frac{\sqrt{2}}{2})^T$ (Figure 8 right). Choosing four of the equations in (21), it is straightforward to compute in this order

$$a_{0,1} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \; a_{0,-1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \; a_{-1,-1} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}, \; a_{1,-1} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}.$$

If we now set $a_{1,0} = \mathrm{av}[(a_{1,-1}, a_{1,1}), (\frac{1}{2}, \frac{1}{2})]$, then the constraint $a_{0,0} = \mathrm{av}[(a_{-1,0}, a_{1,0}), (\frac{1}{2}, \frac{1}{2})] := \tilde{a}_{0,0}$ is violated. □

Apart from the incompatibility of the equations around $a_{0,0}$, we do not necessarily achieve $\mathcal{C}^1$-continuity for Bézier surfaces of type *I* to *III* even if control points satisfying all above conditions can be found. This observation is illustrated in the top line of Figure 10.

Proposition 17. *The conditions* (19) *are not sufficient to ensure* $\mathcal{C}^1$*-continuity of the patched Bézier surface* (18) *of type I, II, or III.*

Proof. **Types *I* and *III*.** We first give a counterexample for types *I* and *III*. Consider $K = 1$ in which case β_K^I and β_K^{III} coincide and each Bézier surface is just obtained by weighted geodesic averaging between four control points. Let us write $b_{ij}^{\mathrm{l}} \equiv b_{i-1,j}^{\mathrm{r}} \equiv b_{ij}$ and choose $b_{ij} = (i,j)$ in Euclidean space $\mathbb{R}^2$. Let the metric be slightly decreased somewhere along the connecting line between b_{00} and $y = (1, \frac{1}{2})$. That way, the geodesic from b_{00} to y is still a straight curve in $\mathbb{R}^2$ but has a shorter length than in the Euclidean space (Figure 9, left). Denote the patched Bézier spline surface by $\mathfrak{B}(t_1, t_2)$ and consider the point $(t_1, t_2) = (1, \frac{1}{2})$. In the Bézier surface patch defined by $b_{00}, b_{01}, b_{10}, b_{11}$ and the one defined by $b_{10}, b_{11}, b_{20}, b_{21}$, we have respectively

$$\begin{aligned} \mathfrak{B}(t_1,t_2) &= \mathrm{av}[(b_{00}, b_{01}, b_{10}, b_{11}), ((1-t_1)(1-t_2), (1-t_1)t_2, t_1(1-t_2), t_1t_2)], \\ \mathfrak{B}(t_1,t_2) &= \mathrm{av}[(b_{10}, b_{11}, b_{20}, b_{21}), ((2-t_1)(1-t_2), (2-t_1)t_2, (t_1-1)(1-t_2), (t_1-1)t_2)]. \end{aligned}$$

Obviously, $b := \mathfrak{B}(1, \frac{1}{2}) = (1, \frac{1}{2})$, and the optimality conditions for both averages are [16, Thm. 1.2]

$$\begin{aligned} 0 &= (1-t_1)(1-t_2)\log_b b_{00} + (1-t_1)t_2 \log_b b_{01} + t_1(1-t_2)\log_b b_{10} + t_1t_2 \log_b b_{11} =: F(t_1,t_2,b), \\ 0 &= (2-t_1)(1-t_2)\log_b b_{10} + (2-t_1)t_2 \log_b b_{11} + (t_1-1)(1-t_2)\log_b b_{20} + (t_1-1)t_2 \log_b b_{21}, \end{aligned}$$

where $\log_b b_{i,j} = b_{i,j} - b$ except for $\log_b b_{0,0} = \zeta(b_{0,0} - b)$ with some $\zeta \in (0,1)$. The latter equation can be solved to yield $b = \mathfrak{B}(t_1,t_2) = (2-t_1)(1-t_2)b_{1,0} + (2-t_1)t_2 b_{1,1} + (t_1 -$

$1)(1-t_2)b_{2,0}+(t_1-1)t_2b_{2,1}$ so that the right derivative $\frac{\partial\mathfrak{B}}{\partial t_1}$ at $(t_1,t_2)=(1,\frac{1}{2})$ equals $(1,0)$. The left derivative can be obtained via the implicit function theorem as

$$\begin{aligned}\frac{\partial\mathfrak{B}}{\partial t_1}&=-(D_bF(1,\tfrac{1}{2},b))^{-1}D_{t_1}F(1,\tfrac{1}{2},b)=D_{t_1}F(1,\tfrac{1}{2},b)\\&\qquad\qquad=-\frac{\log_b b_{00}+\log_b b_{01}-\log_b b_{10}-\log_b b_{11}}{2}=\zeta\begin{pmatrix}\frac{1}{2}\\\frac{1}{4}\end{pmatrix}+\begin{pmatrix}\frac{1}{2}\\-\frac{1}{4}\end{pmatrix}\neq\begin{pmatrix}1\\0\end{pmatrix}.\end{aligned}$$

Here we have used that around $b=(1,\frac{1}{2})$ the function F is given by $F(1,\frac{1}{2},b)=\frac{1}{2}\log_b b_{10}+\frac{1}{2}\log_b b_{11}=\frac{1}{2}(b_{10}-b)+\frac{1}{2}(b_{11}-b)$. Thus, the Bézier surfaces are not patched $\mathcal{C}^1$-continuously, even though all conditions (19) are satisfied.

Type *II*. For a counterexample in case of Bézier surfaces of type *II*, we take again $K=1$ and the same control points as before. This time we consider the Euclidean metric with the modification that it is slightly decreased somewhere in between b_{00} and $(b_{00}+b_{01})/2$: the connecting geodesic from b_{00} to b_{01} is given by $t\mapsto(0,f(t))$ with $f(t)>t$ in the interval $(\frac{1}{2},1)$ (Figure 9 right). We have $\beta_1(t_2;b_{i0},b_{i1})=(i,t_2)$ for $i\neq0$ and $\beta_1(t_2;b_{00},b_{01})=(0,f(t_2))$. Obviously, for $t_1\geq1$ we obtain $\mathfrak{B}(t_1,t_2)=(t_1,t_2)$, while for $t_1\in[0,1]$ and $t_2\in(\frac{1}{2},1)$ we have $\mathfrak{B}(t_1,t_2)=(t_1,(1-t_1)f(t_2)+t_1t_2)$, yielding a discontinuous first derivative at $t_1=1$. □

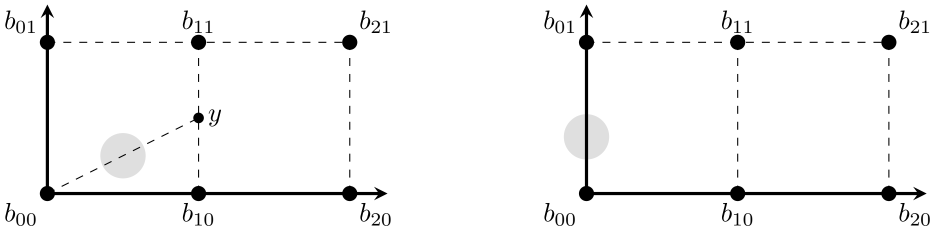


Figure 9: Illustration of the two manifolds from the counterexamples in Proposition 17. A gray region indicates a deviation from the Euclidean metric.

Unfortunately, in case of β_K^{III}, no straightforward remedy can be found to modify de Casteljau's algorithm in a way that is consistent with Bézier surfaces in Euclidean space and, at the same time, allows to have $\mathcal{C}^1$-continuity. For Bézier surfaces of types *I* and *II*, however, one can find such a remedy, as described next.

3.5 $\mathcal{C}^1$-patching solution

We propose now a way to overcome the different problems presented in the previous section. For notational simplicity, let $b_{ij}^{mn}\in\mathcal{M}$ be the ij^{th} control point of the mn^{th} Bézier surface, where $i,j\in\{0,...,K\}$ and $(m,n)\in\{0,...,M\}\times\{0,...,N\}$. We intend to patch the Bézier surfaces together to a $\mathcal{C}^1$-continuous spline surface according to

$$\begin{aligned}&\mathfrak{B}^Y:[0,M]\times[0,N]\to\mathcal{M}:\\&\quad(t_1,t_2)\mapsto\beta_K^Y(t_1-m,t_2-n;(b_{ij}^{mn})_{i,j=0,\dots,K})\quad\text{on }(t_1,t_2)\in[m,m+1]\times[n,n+1]\end{aligned}\tag{22}$$

for $Y=I,II$. As before, we will also allow indices outside the usual range, using the identification

$$b_{-1,j}^{mn}=b_{K-1,j}^{m-1,n},\quad b_{K+1,j}^{mn}=b_{1,j}^{m+1,n},\quad b_{j,-1}^{mn}=b_{j,K-1}^{m,n-1},\quad b_{j,K+1}^{mn}=b_{j,1}^{m,n+1}.$$

Let us additionally define fictional points beyond the domain boundary as

$$\begin{aligned}
&b^{0,n}_{-1,j} = \exp_{b^{0,n}_{0,j}}\left(-\log_{b^{0,n}_{0,j}}\left(b^{0,n}_{1,j}\right)\right), \quad b^{M,n}_{K+1,j} = \exp_{b^{M,n}_{K,j}}\left(-\log_{b^{M,n}_{K,j}}\left(b^{M,n}_{K-1,j}\right)\right), \\
&\hspace{20em} j = 0, \dots, K,\ n = 0, \dots, N, \\
&b^{m,0}_{j,-1} = \exp_{b^{m,0}_{j,0}}\left(-\log_{b^{m,0}_{j,0}}\left(b^{m,0}_{j,1}\right)\right), \quad b^{m,N}_{j,K+1} = \exp_{b^{m,N}_{j,K}}\left(-\log_{b^{m,N}_{j,K}}\left(b^{m,N}_{j,K-1}\right)\right), \\
&\hspace{16em} j = -1, \dots, K+1,\ m = 0, \dots, M.
\end{aligned}$$

In fact, in Euclidean space the conditions

$$b^{mn}_{K,j} = \frac{b^{mn}_{K-1,j} + b^{mn}_{K+1,j}}{2}$$

imply that the control points $b^{mn}_{K,j}$ can be ignored altogether; indeed, in (5) one may simply replace any $b^{mn}_{K,j}$ by $\frac{b^{mn}_{K-1,j}+b^{mn}_{K+1,j}}{2}$. The analogous holds true for the control points $b^{mn}_{i,K}$, $b^{mn}_{i,0}$, and $b^{mn}_{0,j}$. Note that this trick will restore $\mathcal{C}^1$-continuity in the Riemannian setting as detailed below, but it may also be interesting in the Euclidean setting since it allows one to neglect conditions (19) and obtain a differentiable piecewise-Bézier surface nevertheless. Following this idea we obtain a generalized Bézier surface as a weighted average of b^{mn}_{ij}, $i, j \in \mathcal{I}$, with $\mathcal{I} = \{-1, 1, 2, \dots, K-1, K+1\}$ as

$$\beta^{I}_K(t_1, t_2; (b^{mn}_{ij})_{i,j=0,\dots,K}) = \operatorname{av}\left[(b^{mn}_{ij})_{i,j\in\mathcal{I}}, (w_i(t_1)w_j(t_2))_{i,j\in\mathcal{I}}\right] \tag{23}$$

with weights

$$w_i(t) = \begin{cases} \frac{1}{2}B_{0K}(t) & \text{if } i = -1, \\ B_{1K}(t) + \frac{1}{2}B_{0K}(t) & \text{if } i = 1, \\ B_{iK}(t) & \text{if } i = 2, \dots, K-2, \\ B_{K-1,K}(t) + \frac{1}{2}B_{KK}(t) & \text{if } i = K-1, \\ \frac{1}{2}B_{KK}(t) & \text{if } i = K+1, \\ 0 & \text{if } i \in \{0, K\}. \end{cases}$$

Under the same conditions as earlier, every Bézier surface patch is a smooth function into the multigeodesically convex hull of the control points.

Similarly, in (6) the one-dimensional Bézier curves can be computed just based on b^{mn}_{ij}, $i, j \in \mathcal{I}$, *i.e.*, we define

$$\beta^{II}_K(t_1, t_2; (b^{mn}_{ij})_{i,j=0,\dots,K}) = \beta_K(t_1; (\beta_K(t_2; (b^{mn}_{ij})_{j=0,\dots,K}))_{i=0,\dots,K}), \tag{24}$$

where the one-dimensional Bézier curves are either computed as

$$\beta_K(t; (b_j)_{j=0,\dots,K}) = \operatorname{av}[(b_j)_{j=\mathcal{I}}, (w_j(t))_{j=0,\dots,K}],$$

or via the alternative modification of de Casteljau's algorithm where $\beta_K(t; (b_k)_{k=0,\dots,K}) = \mathfrak{b}^K_0$ with

$$\begin{aligned}
\mathfrak{b}^0_0 &= \operatorname{av}[(b_{-1}, b_1), (\tfrac{1}{2}, \tfrac{1}{2})], \\
\mathfrak{b}^0_j &= b_j, & j &= 1, \dots, K-1, \\
\mathfrak{b}^0_K &= \operatorname{av}[(b_{K-1}, b_{K+1}), (\tfrac{1}{2}, \tfrac{1}{2})], \\
\mathfrak{b}^k_j &= \operatorname{av}[(\mathfrak{b}^{k-1}_j, \mathfrak{b}^{k-1}_{j+1}), (1-t, t)], & k &= 1, \dots, K,\ j = 0, \dots, K-k.
\end{aligned}$$

In Euclidean space, all definitions are equivalent to the original Bézier surface as long as the control points satisfy the conditions for $\mathcal{C}^1$-continuity. In the manifold case, where the conditions (21) can at most approximately be satisfied, the above definitions lead to $\mathcal{C}^1$-continuity as shown in Figure 10 and the following theorem. Let us emphasize once more that the $\mathcal{C}^1$-smoothness holds regardless of whether or how well any conditions of type (19) are satisfied.

Theorem 18. *The patched Bézier surface spline* (22) *with β_K^I or β_K^{II} defined as above is $\mathcal{C}^1$-continuous.*

Proof. Each single patch is $\mathcal{C}^1$-continuous by the proof of Corollary 13. Also, $\mathcal{C}^0$-continuity follows as before, so it remains to show that the normal derivatives at all interfaces between two adjacent Bézier patches coincide.

Consider the interface between patch $(0,0)$ and $(1,0)$ at $t_1 = 1$, $t_2 \in [0,1]$. The proof for the other interfaces works analogously.

In case of type I Bézier patches, we have

$$\begin{aligned} b := \mathfrak{B}^I(t_1 = 1, t_2) &= \operatorname{av}\left[(b_{ij}^{0,0})_{i,j\in\mathcal{I}}, (w_i(1)w_j(t_2))_{i,j\in\mathcal{I}}\right] \\ &= \operatorname{av}\left[(b_{ij}^{1,0})_{i,j\in\mathcal{I}}, (w_i(0)w_j(t_2))_{i,j\in\mathcal{I}}\right]. \end{aligned}$$

The optimality conditions for both averages read [16, Thm. 1.2]

$$\begin{aligned} 0 &= \sum_{i,j\in\mathcal{I}} w_i(t_1)w_j(t_2)\log_b b_{ij}^{0,0} && =: F_1(t_1,t_2,b), \\ 0 &= \sum_{i,j\in\mathcal{I}} w_i(t_1-1)w_j(t_2)\log_b b_{ij}^{1,0} && =: F_2(t_1,t_2,b). \end{aligned}$$

By the implicit function theorem, the left and right derivatives of $\mathfrak{B}^I$ with respect to t_1 at $(1, t_2)$ are given as

$$\begin{aligned} \left.\frac{\partial\mathfrak{B}^I}{\partial t_1}\right|_{(1^-,t_2)} &= -(D_bF_1(1,t_2,b))^{-1}D_{t_1}F_1(1,t_2,b) \\ &= -(D_bF_1(1,t_2,b))^{-1}\left(\frac{K}{2}\sum_{j\in\mathcal{I}} w_j(t_2)(\log_b b_{K+1,j}^{0,0} - \log_b b_{K-1,j}^{0,0})\right), \\ \left.\frac{\partial\mathfrak{B}^I}{\partial t_1}\right|_{(1^+,t_2)} &= -(D_bF_2(1,t_2,b))^{-1}D_{t_1}F_2(1,t_2,b) \\ &= -(D_bF_2(1,t_2,b))^{-1}\left(\frac{K}{2}\sum_{j\in\mathcal{I}} w_j(t_2)(\log_b b_{1,j}^{1,0} - \log_b b_{-1,j}^{1,0})\right). \end{aligned}$$

Noting that $F_1(1,t_2,b) = F_2(1,t_2,b)$ for all $b \in \mathcal{M}$ and the definition of the $b_{ij}^{0,0}$ for $i \notin \{0,\dots,K\}$ we see that $\left.\frac{\partial\mathfrak{B}^I}{\partial t_1}\right|_{(1^-,t_2)} = \left.\frac{\partial\mathfrak{B}^I}{\partial t_1}\right|_{(1^+,t_2)}$.

In case of type *II* Bézier patches we note that

$$\begin{aligned} \tfrac{\partial}{\partial t}\beta_K^{II}(t=0;(b_j)_{j=0,\dots,K}) &= \log_b b_1 && \text{for } b = \operatorname{av}[(b_{-1},b_1),(\tfrac12,\tfrac12)], \\ \tfrac{\partial}{\partial t}\beta_K^{II}(t=1;(b_j)_{j=0,\dots,K}) &= \log_b b_{K+1} && \text{for } b = \operatorname{av}[(b_{K-1},b_{K+1}),(\tfrac12,\tfrac12)]. \end{aligned}$$

Thus we have

$$\begin{aligned} \left.\frac{\partial\mathfrak{B}^{II}}{\partial t_1}\right|_{(1^-,t_2)} &= \log_{b(t_2)}\beta_K^{II}(t_2;(b_{K+1,j}^{0,0})_{j=0,\dots,K}), \\ \left.\frac{\partial\mathfrak{B}^{II}}{\partial t_1}\right|_{(1^+,t_2)} &= \log_{b(t_2)}\beta_K^{II}(t_2;(b_{1,j}^{1,0})_{j=0,\dots,K}) \end{aligned}$$

for $b(t_2) = \operatorname{av}[(\beta_K^{II}(t2;(b_{K-1,j}^{0,0})_{j=0,\dots,K}), \beta_K^{II}(t2;(b_{K+1,j}^{0,0})_{j=0,\dots,K})),(\frac12,\frac12)]$. Again, the derivatives from either side coincide. □

In the next section we propose a way to choose the control points such that the interpolating surface has the least squared acceleration when the manifold reduces to the Euclidean space.

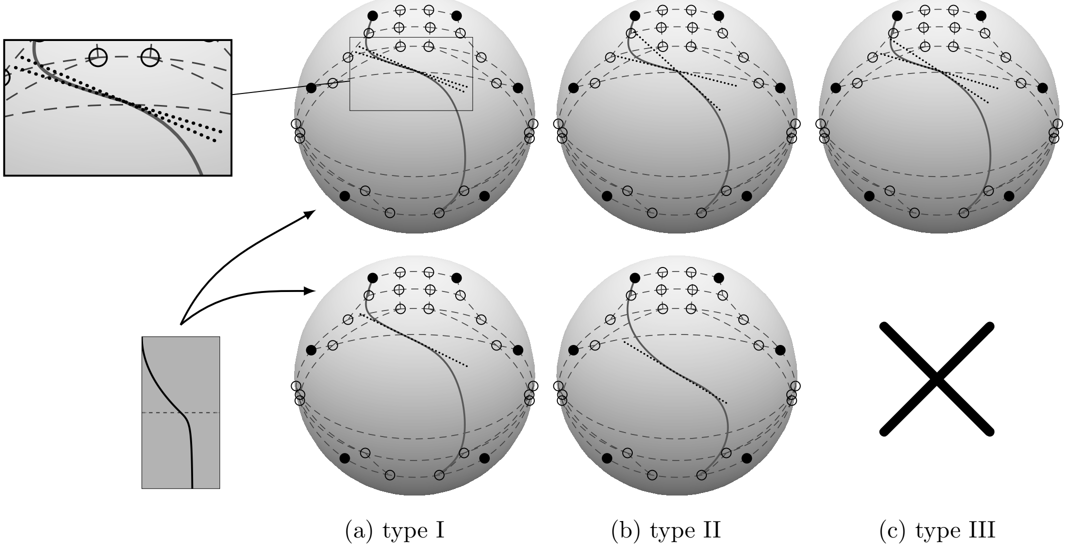


Figure 10: The conditions (19) do not suffice to ensure differentiability (top, visualized using a piecewise Bézier surface composed of two patches) while the remedy from Theorem 18 permits to construct $\mathcal{C}^1$ surfaces of type *I* and type *II* (bottom). The visualiszation is analogous to Figure 4 left; the dotted lines show tangents to the curve from either side of the interface between the two Bézier patches.

4 Control point generation for 2D piecewise cubic Bézier interpolation on manifolds

Given data points

$$p_{mn} \in \mathcal{M}, \quad (m,n) \in \{0,\dots,M\} \times \{0,\dots,N\},$$

we would like to interpolate those by a smooth surface $\mathfrak{B} : [0,M] \times [0,N] \to \mathcal{M}$ with $\mathfrak{B}(m,n) = p_{mn}$, which consists of $\mathcal{C}^1$-continuously patched cubic Bézier surfaces on each domain $[m,m+1] \times [n,n+1]$ as in (22). To this end we need to generate appropriate control points

$$b_{ij}^{mn} \text{ for } m,n \in \{0,\dots,M-1\} \times \{0,\dots,N-1\} \text{ and } i,j = 0,\dots,3.$$

The control points must respect the interpolation constraints for $m = 0,\dots,M-1$, $n = 0,\dots,N-1$,

$$b_{0,0}^{mn} = p_{m,n}, \quad b_{3,0}^{mn} = p_{m+1,n}, \quad b_{0,3}^{mn} = p_{m,n+1}, \quad b_{3,3}^{mn} = p_{m+1,n+1}. \tag{25}$$

as well as the $\mathcal{C}^0$-patching constraints of Theorem 14,

$$b_{3,j}^{m,n} = b_{0,j}^{m+1,n} \quad \text{and} \quad b_{j,3}^{m,n} = b_{j,0}^{m,n+1} \tag{26}$$

for $j = 0,\dots,3$. Furthermore, the resulting spline surface shall be $\mathcal{C}^1$-smooth and must thus be generated by the approach of Theorem 18.

To make the interpolating surface as nice as possible, we would like to optimize the position of the control points such that the mean squared second derivative of the parameterized spline surface is minimized. This, however, is a highly complicated nonlinear optimization problem. Instead, we consider a much simpler approximation: we formulate the problem of finding the optimal control points in Euclidean space (similarly to [11]), in which case the optimization problem reduces to the solution of a linear system. We then transfer this linear system to the manifold case. The following subsections elaborate this idea.

4.1 Variational formulation of control point generation in $\mathbb{R}^n$

In the Euclidean space $\mathbb{R}^n$, we would like to minimize the objective function

$$F[\mathfrak{B}] = \int_{[0,M]\times[0,N]} \left\| \frac{\partial^2 \mathfrak{B}}{\partial(t_1,t_2)} \right\|_{\mathrm{F}}^2 \mathrm{d}(t_1,t_2) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \int_{[0,1]\times[0,1]} \left\| \frac{\partial^2 \beta_3^{mn}}{\partial(t_1,t_2)} \right\|_{\mathrm{F}}^2 \mathrm{d}(t_1,t_2), \quad (27)$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm on $(2\times 2\times n)$-tensors, $\frac{\partial^2 \cdot}{\partial(t_1,t_2)}$ is the Hessian operator for any bivariate function and β_3^{mn} is the cubic Bézier surface defined on the patch (m,n) and based on the control points b_{ij}^{mn}. For notational simplicity, we will just denote it by β when no confusion is possible. Note that the constraints (25) will automatically ensure the interpolation of the data points.

Expressing each single Bézier surface patch with the help of Bernstein polynomials according to (5), the objective function turns into a quadratic function in the control points b_{ij}^{mn},

$$F[\mathfrak{B}] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \hat{F}[\beta^{mn}], \quad (28)$$

where the energy $\hat{F}$ for a Bézier surface with control points b_{ij}^{mn}, $i,j\in\{0,\dots,3\}$, is defined as

$$\hat{F}[\beta^{mn}] = \sum_{i,j,o,p=0}^{3} \alpha_{ijop} b_{ij}^{mn}\cdot b_{op}^{mn} \quad (29)$$

with $\cdot$ indicating the Euclidean dot product. Denoting the Frobenius inner product on 2×2-matrices by $A : B = \sum_{i,j=1}^{2} A_{ij}B_{ij}$, the coefficients α_{ijop} in the above energy are given by

$$\alpha_{ijop} = \int_{[0,1]^2} \left[\frac{\partial^2 B_{i3}(t_1)B_{j3}(t_2)}{\partial(t_1,t_2)}\right] : \left[\frac{\partial^2 B_{o3}(t_1)B_{p3}(t_2)}{\partial(t_1,t_2)}\right] \mathrm{d}(t_1,t_2)\,, \quad (30)$$

where the explicit Hessian of the Bernstein polynomial products can be expressed as

$$\frac{\partial^2 B_{i3}(t_1)B_{j3}(t_2)}{\partial(t_1,t_2)} = \begin{pmatrix} \frac{\partial^2 B_{i3}(t_1)}{\partial^2 t_1} B_{j3}(t_2) & \frac{\partial B_{i3}(t_1)}{\partial t_1}\frac{\partial B_{j3}(t_2)}{\partial t_2} \\ \frac{\partial B_{i3}(t_1)}{\partial t_1}\frac{\partial B_{j3}(t_2)}{\partial t_2} & B_{i3}(t_1)\frac{\partial^2 B_{j3}(t_1)}{\partial^2 t_2} \end{pmatrix}. \quad (31)$$

Note that the coefficients α_{ijop} can be readily computed analytically and are independent of the configuration.

To later be able to transfer this formulation to the manifold setting, we would like to express every control point of a given patch as its difference with the four interpolation points of the patch. These differences will be later translated into Riemannian logarithms. Since the objective function $F[\mathfrak{B}]$ only contains derivatives, the contributions $\hat{F}[\beta]$ of its single Bézier patches are invariant under a uniform translation of the control points. Hence

$$\hat{F}[\beta^{mn}] = \frac{1}{4}\sum_{r,s\in\{0,1\}}\sum_{i,j,o,p=0}^{3} \alpha_{ijop} v_{ij}^{mn}(r,s)\cdot v_{op}^{mn}(r,s), \quad (32)$$

where we introduced the auxiliary variables for the differences (see Figure 11)

$$v_{ij}^{mn}(r,s) = b_{ij}^{mn} - p_{m+r,n+s} \quad (33)$$

for $i,j=0,\dots,3$, $r,s=0,1$, $m=0,\dots,M-1$ and $n=0,\dots,N-1$. Note that for symmetry reasons we shifted the control points by each corner of the corresponding patch and then took the average of the four energy values resulting from those four shifts.

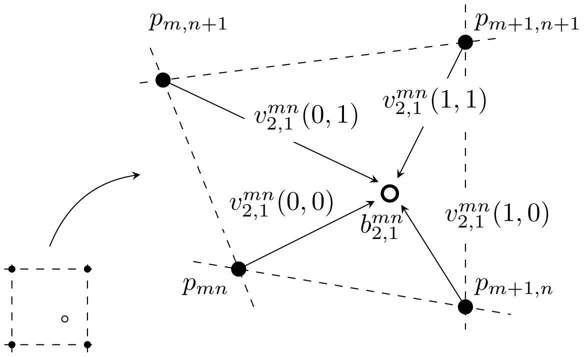


Figure 11: Geometric interpretation of variables $v^{mn}_{ij}(r,s)$.

To summarize, the total energy can be represented as

$$F[\mathfrak{B}] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{r,s\in\{0,1\}}\sum_{i,j=0}^{3}(L(V))^{mn}_{ij,rs}\cdot v^{mn}_{ij}(r,s) \tag{34}$$

with $V = (v^{mn}_{ij}(r,s))^{(m,n)\in\{0,\ldots,M-1\}\times\{0,\ldots,N-1\}}_{i,j\in\{0,\ldots,3\},\ r,s\in\{0,1\}}$ and the linear operator L

$$(L(V))^{mn}_{ij,rs} = \frac{1}{4}\sum_{o,p=0}^{3}\alpha_{ijop}v^{mn}_{op}(r,s). \tag{35}$$

This energy has to be minimized for the control points b^{mn}_{ij} or equivalently the $v^{mn}_{ij}(r,s)$ under interpolation, continuity, and smoothness constraints (25), (26), and (21).

4.2 System reduction by constraint elimination

Before minimizing (34), we would like to eliminate all constraints (25), (26), and (21) in order to reduce the number of degrees of freedom and to reach a simpler unconstrained minimization problem. To this end, without loss of generality, we consider as independent the control points

$$b^{mn}_{kl} \text{ with } (k,l)\in Q = \{(1,0),(0,1),(1,1)\},\ (m,n)\in D = \{0,\ldots,M\}\times\{0,\ldots,N\}.$$

Conditions (25), (26), and (21) then uniquely determine all the remaining control points, *e.g.*, $b^{mn}_{-1,1} = 2b^{mn}_{0,1} - b^{mn}_{1,1}$. In the above, we again extended our notation to allow control point indices outside $\{0,\ldots,3\}$ (*cf.* Figure 8 left), with the interpretation

$$b^{mn}_{kl} = b^{m,n-1}_{k,3+l} = b^{m-1,n}_{3+k,l} = b^{m-1,n-1}_{3+k,3+l}$$

for $k,l\in\{-1,0,1\}$, where points b^{mn}_{ij} with $(m,n)\notin\{0,\ldots,M-1\}\times\{0,\ldots,N-1\}$ are just ficticious additional control points.

Equivalently to the b^{mn}_{kl}, $(k,l)\in Q$, $(m,n)\in D$, we consider the translated control points

$$u^{mn}_{kl} = b^{mn}_{kl} - p_{mn}, \qquad (k,l)\in Q,\ (m,n)\in D, \tag{36}$$

as our independent arguments. Those will be the arguments we optimize. In the following, we will express the energy variables $v^{mn}_{ij}(r,s)$ in terms of these u^{mn}_{kl}. This requires two operators:

1. The linear operator S generates u^{mn}_{kl} for $(k,l)\in\{-1,0,1\}^2\setminus Q$ from the conditions of

$\mathcal{C}^1$-continuity (21) (*cf.* Figure 12),

$$S : (u_{kl}^{mn})_{(k,l)\in Q}^{(m,n)\in D} \mapsto (u_{kl}^{mn})_{k,l\in\{-1,0,1\}}^{(m,n)\in D} : \begin{cases} u_{-1,1}^{mn} &= 2u_{0,1}^{mn} - u_{1,1}^{mn}, \\ u_{0,0}^{mn} &= 0, \\ u_{-1,0}^{mn} &= -u_{1,0}^{mn}, \\ u_{-1,-1}^{mn} &= -u_{1,1}^{mn}, \\ u_{0,-1}^{mn} &= -u_{0,1}^{mn}, \\ u_{1,-1}^{mn} &= 2u_{1,0}^{mn} - u_{1,1}^{mn}, \\ u_{kl}^{mn} &= u_{kl}^{mn} \text{ for } (k,l)\in Q. \end{cases} \tag{37}$$

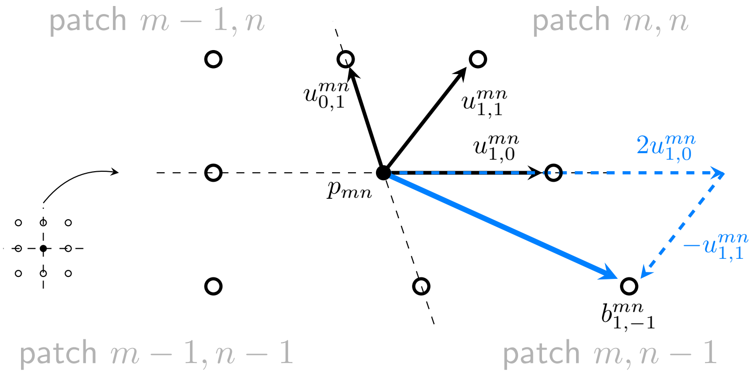


Figure 12: Geometric interpretation of operator S.

2. The operator $\tilde{T}$ maps the set of vectors u_{kl}^{mn} onto the set of vectors $v_{ij}^{mn}(r,s)$ by exploiting the relation

$$v_{ij}^{mn}(r,s) = u_{kl}^{\tilde{m}\tilde{n}} + (p_{\tilde{m}\tilde{n}} - p_{m+r,n+s}),$$

where $\tilde{m}$, $\tilde{n}$, k, and l satisfy

$$(\tilde{m},\tilde{n}) = (m+a_i, n+a_j) \quad \text{and} \quad (k,l) = (i-3a_i, j-3a_j) \quad \text{for} \quad a_i = \begin{cases} 0 & \text{if } i\in\{0,1\}, \\ 1 & \text{if } i\in\{2,3\}. \end{cases}$$

To this end we first introduce the notation

$$w_{ij}^{mn}(r,s) = u_{kl}^{\tilde{m}\tilde{n}}, \tag{38}$$
$$z_{ij}^{mn}(r,s) = p_{\tilde{m}\tilde{n}} - p_{m+r,n+s}, \tag{39}$$

which seems cumbersome at first but is needed for an easy transfer to the manifold setting. Abbreviating

$$U \equiv \left(u_{kl}^{mn}\right)_{k,l\in\{-1,0,1\}}^{(m,n)\in D}, \qquad V \equiv \left(v_{ij}^{mn}(r,s)\right)_{\substack{i,j=0,\dots,3,\\ r,s\in\{0,1\}}}^{(m,n)\in D},$$
$$W \equiv \left(w_{ij}^{mn}(r,s)\right)_{\substack{i,j=0,\dots,3,\\ r,s\in\{0,1\}}}^{(m,n)\in D}, \qquad Z \equiv \left(z_{ij}^{mn}(r,s)\right)_{\substack{i,j=0,\dots,3,\\ r,s\in\{0,1\}}}^{(m,n)\in D},$$

and introducing the linear operator

$$T : U \mapsto W \tag{40}$$

we finally obtain

$$\tilde{T} : U \mapsto V = T(U) + Z. \tag{41}$$

Note that the operator T can be interpreted to operate on each patch (m, n) separately as follows (this interpretation forms the basis for the transfer to the manifold setting): T translates any vector $u_{kl}^{\tilde{m}\tilde{n}}$, which belongs to a control point of the patch, from its base point $p_{\tilde{m}\tilde{n}}$ to all four patch corners $p_{m+r,n+s}$, $(r, s) \in \{0, 1\}^2$, resulting in the four new vectors $w_{ij}^{mn}(r, s)$, $r, s \in \{0, 1\}$. Of course, in Euclidean space this translation is trivial, however, it will turn into a nontrivial parallel transport in the manifold setting. The map $\tilde{T}$ is illustrated in Figure 13.

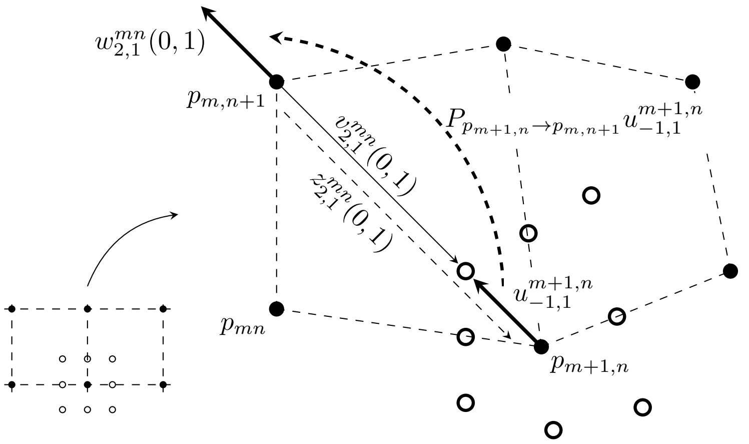


Figure 13: Geometric interpretation of operator $\tilde{T}$. The variables $v_{ij}^{mn}(r,s)$ (*cf.* Figure 11) are constructed as follow: (i) a vector u_{kl}^{mn} in p_{mn} is transported to another interpolation point p, and (ii) to this vector one adds the difference between p and p_{mn}.

The total energy (34) of Bézier curves in Euclidean space can thus be rewritten as

$$F[\mathfrak{B}] = \frac{1}{4} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{r,s \in \{0,1\}} \sum_{i,j=0}^{3} (L\tilde{T}S\tilde{U})_{ij,rs}^{mn} \cdot (\tilde{T}S\tilde{U})_{ij,rs}^{mn} \qquad \text{for } \tilde{U} = (u_{kl}^{mn})_{(k,l) \in Q}^{(m,n) \in D}$$

and is minimized by

$$\tilde{U}_{\text{opt}} = -(S^*T^*LTS)^{-1}(S^*T^*LZ)\,, \tag{42}$$

where a superscript asterisk denotes the adjoint operator (note that L is self-adjoint). Indeed, letting $(\cdot;\cdot)$ denote the natural inner product on Cartesian products of vector spaces (such that for instance $F[\mathfrak{B}] = (L\tilde{T}S\tilde{U}; \tilde{T}S\tilde{U}) = (L(TS\tilde{U} + Z); TS\tilde{U} + Z))$, this is easily seen from the first-order optimality condition

$$\begin{aligned} 0 = (LTS\Phi; TS\tilde{U}_{\text{opt}} + Z) &+ (LTS\tilde{U}_{\text{opt}} + LZ; TS\Phi) \\ &= 2(LTS\tilde{U}_{\text{opt}} + LZ; TS\Phi) = 2(S^*T^*LTS\tilde{U}_{\text{opt}} + S^*T^*LZ; \Phi) \end{aligned}$$

for all variations Φ of $\tilde{U}_{\text{opt}}$.

4.3 Transfer to the manifold setting

It remains to transfer the different operators and the final formula (42) for control point generation to a Riemannian manifold setting.

- The u_{kl}^{mn} were defined in (36) as the difference between two Euclidean points. Their generalization to the Riemannian setting is given by

  $$u_{kl}^{mn} = \log_{p_{mn}} \left(b_{kl}^{mn}\right),$$

  for $k, l \in Q$, $m = 0, \ldots, M-1$ and $n = 0, \ldots, N-1$. This means that we will actually optimize over tangent vectors u_{kl}^{mn} to the manifold and only afterwards convert them into control points b_{kl}^{mn}.

- The operator L defined in (35) can now be interpreted as an operator from B into itself, where B is the Cartesian product of tangent spaces

  $$B = \bigtimes_{\substack{i,j=0,\ldots,3 \\ r,s\in\{0,1\} \\ m=0,\ldots,M-1 \\ n=0,\ldots,N-1}} T_{p_{m+r,n+s}}\mathcal{M}.$$

- The operator S in (37) is now considered as an operator on tangent spaces.

- Formula (39), which defines the components of Z, is generalized to the manifold setting as

  $$z_{ij}^{mn}(r,s) = \log_{p_{m+r,n+s}} \left(p_{\tilde{m}\tilde{n}}\right). \tag{43}$$

- Operator T from (40) is redefined as the parallel transport of the variables u_{kl}^{mn} to the corners of the corresponding patch,

  $$T : \left(u_{kl}^{mn}\right)_{k,l\in\{-1,0,1\}}^{(m,n)\in D} \mapsto \left(w_{ij}^{mn}(r,s)\right)_{\substack{i,j=0,\ldots,3 \\ r,s\in\{0,1\}}}^{(m,n)\in D}, \quad w_{ij}^{mn}(r,s) = P_{p_{\tilde{m}\tilde{n}} \to p_{m+r,n+s}} u_{kl}^{\tilde{m}\tilde{n}},$$

  where $P_{x\to y}$ denotes the Riemannian parallel transport from $x \in \mathcal{M}$ to $y \in \mathcal{M}$. We used the same notation as in (38).

- The operator $\tilde{T}$ from (41) is transferred to the manifold setting using the manifold versions of T and Z.

- The adjoint operators S^* and T^* are given by

  $$S^* : (u_{kl}^{mn})_{k,l\in\{-1,0,1\}}^{(m,n)\in D} \mapsto (u_{kl}^{mn})_{(k,l)\in Q}^{(m,n)\in D} : \begin{cases} u_{1,0}^{mn} & = & u_{1,0}^{mn} - u_{-1,0}^{mn} + 2u_{1,-1}^{mn}, \\ u_{1,1}^{mn} & = & u_{1,1}^{mn} - u_{-1,1}^{mn} - u_{-1,-1}^{mn} - u_{1,-1}^{mn}, \\ u_{0,1}^{mn} & = & u_{0,1}^{mn} - u_{0,-1}^{mn} + 2u_{-1,1}^{mn}, \end{cases}$$

  $$T^* : \left(w_{ij}^{mn}(r,s)\right)_{\substack{i,j=0,\ldots,3 \\ r,s\in\{0,1\}}}^{(m,n)\in D} \mapsto \left(u_{kl}^{mn}\right)_{k,l\in\{-1,0,1\}}^{(m,n)\in D},$$

  $$u_{kl}^{\hat{m}\hat{n}} = \sum_{\substack{r,s\in\{0,1\} \\ m\in\hat{m}+A_k \\ n\in\hat{n}+A_l}} P_{p_{m+r,n+s}\to p_{\hat{m}\hat{n}}} w_{k+3(\hat{m}-m),l+3(\hat{n}-n)}^{mn}(r,s),$$

  where $A_{-1} = \{-1\}$, $A_0 = \{-1, 0\}$, and $A_1 = \{0\}$.

The algorithm for generating the control points on a Riemannian manifold $\mathcal{M}$ now proceeds as follows.

1. Compute $Z = (z_{ij}^{mn}(r,s))_{ijmnrs}$ via (43).

2. Compute S^*T^*LZ and solve (42) for U_{opt} by a conjugate gradient iteration.

3. Compute all u_{kl}^{mn} for $k, l \in \{-1, 0, 1\}$ and $(m,n) \in D$ via SU_{opt}.

4. Compute all control points $b_{kl}^{mn} \in \mathcal{M}$ for $k, l \in \{-1, 1\}$ and $(m, n) \in D$. Note that all other control points are not used in the Bézier spline evaluation (23) or (24) and are thus irrelevant.

In the last step of the algorithm, the computation of the control points has to be performed in a way that ensures $\mathfrak{B}(m, n) = p_{mn}$. This requires a different procedure for Bézier splines of type *I* or *II*.

- For Bézier splines of type *I* we simply use

$$b_{kl}^{mn} = \exp_{p_{mn}}\left(u_{kl}^{mn}\right), \quad k, l \in \{-1, 1\},\ (m, n) \in D,$$

  since this automatically satisfies $\mathfrak{B}^I(m, n) = \mathrm{av}[(b_{-1,-1}^{mn}, b_{-1,1}^{mn}, b_{1,-1}^{mn}, b_{1,1}^{mn}), (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})] = p_{mn}$.

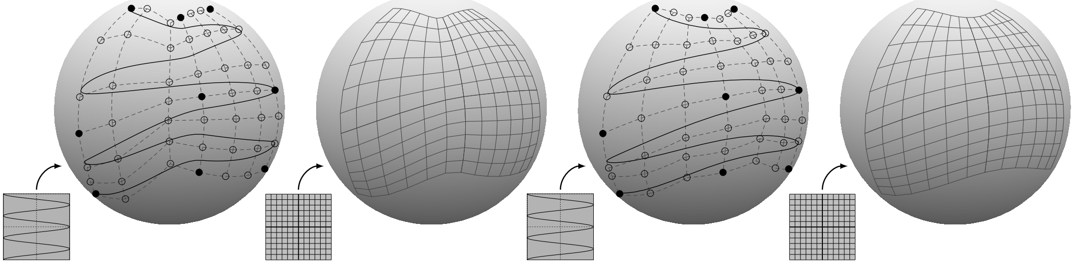
- For Bézier splines of type *II* we set

$$\begin{aligned} b_{k0}^{mn} &= \exp_{p_{mn}}\left(u_{k0}^{mn}\right), \\ b_{kl}^{mn} &= \exp_{b_{k0}^{mn}}\left(P_{p_{mn} \to b_{k0}^{mn}} u_{0l}^{mn}\right), \quad k, l \in \{-1, 1\},\ (m, n) \in D. \end{aligned}$$

  Then,

$$\begin{aligned} \mathfrak{B}^{II}(m, n) &= \mathrm{av}[(\mathrm{av}[(b_{-1,-1}^{mn}, b_{-1,1}^{mn}), (\frac{1}{2}, \frac{1}{2})], \mathrm{av}[(b_{1,-1}^{mn}, b_{1,1}^{mn}), (\frac{1}{2}, \frac{1}{2})]), (\frac{1}{2}, \frac{1}{2})] \\ &= \mathrm{av}[(b_{-1,0}^{mn}, b_{1,0}^{mn}), (\frac{1}{2}, \frac{1}{2})] \\ &= p_{mn}. \end{aligned}$$

An example is provided in Figure 14.

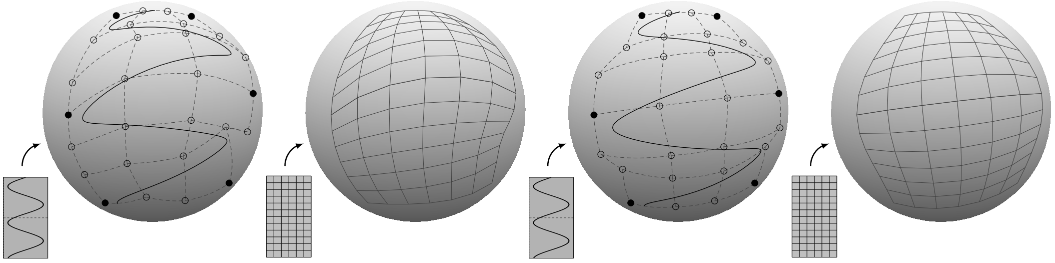


initial control points, $F[\mathfrak{B}] = 8.185$ optimised control points, $F[\mathfrak{B}] = 4.932$

Figure 14: Optimal placement of control points (circles) for given interpolation points (dots) on the sphere obtained by the algorithm described in Section 4. The left graphs show the configuration before optimization (control points b_{kl}^{mn} for $(k, l) \in Q$, $m, n \in \{0, 1\}$ obtained by geodesic averaging between the interpolation points), the right graphs after optimization.

5 Numerical implementation on Riemannian manifolds

A numerical implementation of the approaches from Sections 3 and 4 requires to compute weighted geodesic averages ($\mathrm{av}[\cdot, \cdot]$), exponential maps (exp), logarithms (log) and parallel transports ($P_{\cdot \to \cdot}$) on a Riemannian manifold $\mathcal{M}$. In this section we state how those operations are performed in our numerical examples. First we consider the case of simple manifolds $\mathcal{M}$ in which there are closed formulae for exp, log and $P_{\cdot \to \cdot}$; then we briefly recall the discretization and numerical procedures from [23] for manifolds where this is not the case.



$F[\mathfrak{B}] = 22.16$ $F[\mathfrak{B}] = 15.96$

Figure 15: Instead of transporting the vectors to all four corners of a patch, vectors could also be transported to a single corner in each patch. However, such a modification to the algorithm described in Section 4 will in general yield configurations with a higher energy. The two figures on the left show the result of a modified version of the algorithm, in which all vectors are transported to the bottom left vertex in each patch. The other two figures show the result obtained by the unmodified algorithm.

5.1 Weighted geodesic averaging on manifolds with explicit logarithm

According to Definition 4, Bézier surfaces of type I are obtained as a weighted averaging of the control points of the patch. This weighted geodesic average (*cf.* Definition 1) corresponds to the minimization problem of the form

$$\beta^I(t_1,t_2) = \operatorname*{argmin}_{x\in\mathcal{M}} J(x), \qquad J(x) = \sum_{i=0}^{K}\sum_{j=0}^{K} w_{ij}(t_1,t_2)\mathrm{d}^2(b_{ij},x).$$

On manifolds embedded in the Euclidean space without constraints, this problem can be solved using a Quasi-Newton approach. For other matrix manifolds, different solutions are available, like the iterative Riemannian optimization procedures described in Absil *et al.* [1] (with the exponential map as retraction). These algorithms are implemented in the toolbox Manopt [6]. Equivalently, one could also use standard tools for constrained optimization.

These methods require the evaluation of the objective function $J(x)$ as well as its derivative, which by [16, §1.2.] can be calculated as

$$\frac{\mathrm{d}J}{\mathrm{d}x}(x) = -\sum_{i=0}^{K}\sum_{j=0}^{K} w_{ij}(t_1,t_2)\log_x b_{ij} \in T_x\mathcal{M}\,.$$

5.2 Discrete approximation of Riemannian operators

In many interesting manifolds the standard Riemannian operators can be expressed as closed formulas [1, 5, 22] or can be estimated via retractions as in [6]. However, more complicated manifolds (like the manifold of triangulated shells, *cf.* Figure 16) require a numerical approximation of these operators. To this end we make use of the discrete geodesic calculus from [23]. We briefly recall it in this section.

Discrete distance. Let $\mathcal{M}$ be a smooth Riemannian manifold. Consider a smooth application $W : \mathcal{M}\times\mathcal{M} \to \mathbb{R}$ which approximates the squared Riemannian distance as

$$\mathrm{d}^2(y_1,y_2) = W[y_1,y_2] + \mathcal{O}(\mathrm{d}^3(y_1,y_2)), \quad y_1,y_2\in\mathcal{M}. \tag{44}$$

To ensure the efficiency of the numerical methods, it is important to choose W easy to evaluate. Note also that equation (44) implies straightforwardly that the second derivative of W coincides with the Riemannian metric $g_y = \partial_1^2 W[y,y]$.

Discrete geodesic. Consider now a $(k+1)$-tuple $(y_0, \dots, y_k)$ with $y_j \in \mathcal{M}$ for $j = 0, \dots, k$. This tuple is called *discrete k-path* between y_0 and y_k. Its length L and energy E are defined as

$$L[y_0, \dots, y_k] = \sum_{j=1}^{k} \sqrt{W[y_{j-1}, y_j]}, \tag{45}$$

$$E[y_0, \dots, y_k] = k \sum_{j=1}^{k} W[y_{j-1}, y_j]. \tag{46}$$

The $(k+1)$-tuple minimizing the energy of the discrete k-path is called a *discrete k-geodesic*:

$$\min_{y_1, \dots, y_{k-1} \in \mathcal{M}} E[y_0, y_1, \dots, y_{k-1}, y_k]. \tag{47}$$

In [23] it is shown that discrete geodesics approximate the true continuous geodesics as $k \to \infty$. A discrete k-geodesic is represented in Figure 16.

Discrete weighted average. Now that the discrete k-geodesic is defined, it is possible to express the discrete analog of the weighted geodesic average $\mathrm{av}[(y^1, \dots, y^n), (w_1, \dots, w_n)]$ as the point $y \in \mathcal{M}$ solving

$$\min_{y \in \mathcal{M}} \min_{\substack{y_j^i \in \mathcal{M} \\ j=1,\dots,k-1, \\ i=1,\dots,n}} \sum_{i=1}^{n} w_i E[y, y_1^i, \dots, y_{k-1}^i, y^i]. \tag{48}$$

At optimality, $(y, y_1^i, \dots, y_{k-1}^i, y^i)$ is the discrete k-geodesic from y to y^i, and $E[y, y_1^i, \dots, y_{k-1}^i, y^i]$ is the discrete approximation of the squared Riemannian distance $\mathrm{d}^2(y, y^i)$.

Discrete logarithm. In the following, we require $\mathcal{M}$ to be identified with a subset of some embedding Banach space B such that, for a discrete k-geodesic $(y_0, \dots, y_k)$ with $y_0 = y_A$ and $y_k = y_B$, the difference $y_1 - y_0$ is well-defined. We define the discrete logarithm $\left(\frac{1}{k}\mathrm{LOG}\right)$ as

$$\left(\tfrac{1}{k}\mathrm{LOG}\right)_{y_A}(y_B) = y_1 - y_0, \tag{49}$$

where $\frac{1}{k}$ is to be interpreted as part of the symbol $\left(\frac{1}{k}\mathrm{LOG}\right)$. Under certain regularity assumptions on the Riemannian metric and the functional W, it can be shown that $k\left(\frac{1}{k}\mathrm{LOG}\right)_{y_A}(y_B)$ tends to $\log_{y_A}(y_B)$ when k tends to infinity [23].

Discrete exponential. We now aim at defining a discrete analog EXP^k of the exponential map. For a discrete k-geodesic $(y_0, \dots, y_k)$, we expect EXP^k to reflect the properties of its

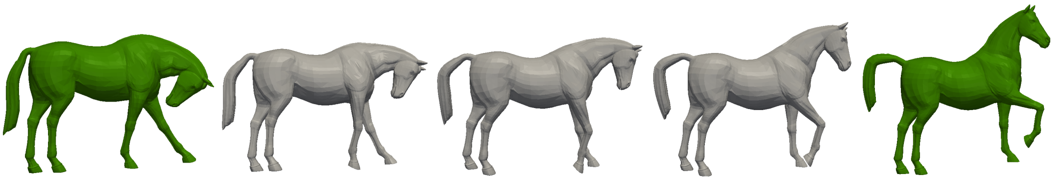


Figure 16: Discrete 4-geodesic between two triangulated shells (*cf.* Section 6). The mesh data used in this figure was made available by Robert Sumner and Jovan Popovic, MIT Computer Graphics Group.

continuous counterpart, *i.e.*, to satisfy $\mathrm{EXP}^k_{y_0}(v) = y_k$ where $v = \left(\frac{1}{k}\mathrm{LOG}\right)_{y_0}(y_k)$. To achieve this, let $v \in B$. Let also

$$\begin{aligned} \mathrm{EXP}^1_y(v) &= (\tfrac{1}{1}\mathrm{LOG})^{-1}_y(v) = y + v, \\ \mathrm{EXP}^2_y(v) &= (\tfrac{1}{2}\mathrm{LOG})^{-1}_y(v). \end{aligned}$$

The *discrete exponential map* is defined recursively as

$$\mathrm{EXP}^k_y(v) = \mathrm{EXP}^2_{\mathrm{EXP}^{k-2}_y(v)}(\tilde{v}) \quad \text{with} \quad \tilde{v} = \mathrm{EXP}^{k-1}_y(v) - \mathrm{EXP}^{k-2}_y(v). \tag{50}$$

Note that EXP^2 is non-trivial and is here simply expressed as the inverse of the discrete logarithm. It remains then to determine a way to compute this object. It follows from the definition of $(\frac{1}{2}\mathrm{LOG})$ that $y_2 = \mathrm{EXP}^2_{y_0}(v)$ satisfies

$$y_0 + v = \mathrm{argmin}_{y \in \mathcal{M}} \left(W[y_0, y] + W[y, y_2] \right).$$

Followingly, y_2 can be obtained by solving the corresponding Euler-Lagrange equation

$$y_2 \in \mathcal{M} : \partial_2 W[y_0, y_0 + v] + \partial_1 W[y_0 + v, y_2] = 0. \tag{51}$$

Discrete parallel transport. To transport a vector along a discrete curve, we use a first order approximation of the parallel transport called *Schild's ladder.* Let $(y_0, \dots, y_n)$ be a discrete curve in $\mathcal{M}$ and $v_0 \in T_{y_0}\mathcal{M}$, the vector to transport from y_0 to y_n. The transported tangent vector v_k at a point y_k, $k \in \{1, \dots, n\}$, can be computed recursively following the algorithm illustrated in Figure 17:

$$\begin{aligned} y^p_{k-1} &= \mathrm{EXP}^1_{y_{k-1}}(v_{k-1}), \\ y^{\mathrm{mid}}_k &= \mathrm{EXP}^1_{y^p_{k-1}}\left((\tfrac{1}{2}\mathrm{LOG})_{y^p_{k-1}}(y_k) \right), \\ y^p_k &= \mathrm{EXP}^2_{y_{k-1}}\left((\tfrac{1}{1}\mathrm{LOG})_{y_{k-1}}(y^{\mathrm{mid}}_k) \right), \\ v_k &= (\tfrac{1}{1}\mathrm{LOG})_{y_k}(y^p_k). \end{aligned}$$

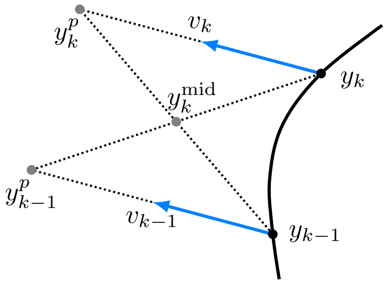


Figure 17: Illustration of one iteration of Schild's ladder, approximating the parallel transport of a vector along a discrete curve $(y_0, \dots, y_n)$.

To find a numerical solution to the problems of the discrete geodesic (47), the discrete averaging (48) and the discrete exponential (51) (all other equations are trivial to solve), we use a Newton's method. These discrete Riemannian operators are used in Sections 6.3 and 6.4. Note that as these objects require to solve an optimization problem, they are usually long to compute.

6 Numerical examples

We finally present here some examples of Bézier spline surfaces computed on specific manifolds such as the sphere, the special orthogonal group or the space of shells. For reasons of computational efficiency (especially on manifolds for which no closed formulae of Riemannian operators are available), the shown examples all represent a Bézier surface of type *II*.

6.1 The sphere $\mathbb{S}^{d-1}$

Table 1 recalls the explicit formulas of the Riemannian operators needed to optimize the control points and evaluate a Bézier spline on the sphere $\mathbb{S}^{d-1}$. A first computational example on the sphere $\mathbb{S}^2$ has already been shown in Figure 14.

As a second example we consider image transfer from the plane onto the sphere. Figure 18 shows a rectangular map of the world, which serves as the parameterization domain of a smooth Bézier spline surface on the sphere. The surface parameterization then provides a one-to-one map between points on the rectangle and points on the sphere, which can be used to map the world image onto the sphere.

Set	$\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d \ : \ x^\top x = 1\}$
Tangent space	$T_x\mathbb{S}^{d-1} = \{v \in \mathbb{R}^d \ : \ x^\top v = 0\}$
Inner product	$\langle v_1, v_2 \rangle_x = v_1^\top v_2$
Distance	$\mathrm{d}(x,y) = \arccos(x^\top y)$
Exponential	$\exp_x(v) = x\cos(\lVert v\rVert) + \frac{v}{\lVert v\rVert}\sin(\lVert v\rVert)$
Logarithm	$\log_x(y) = \frac{(I_d - xx^\top)y}{\sqrt{1-(x^\top y)^2}}\arccos(x^\top y)$
Parallel transport	$P_{x\to y}(v) = -x\sin(\lVert\xi\rVert) + \frac{\xi}{\lVert\xi\rVert}\cos(\lVert\xi\rVert)\xi^\top v + \left(I_d - \frac{\xi\xi^\top}{\lVert\xi\rVert^2}\right)v$, $\xi = \log_x(y)$

Table 1: Riemannian operators for $\mathbb{S}^{d-1}$, *cf.* [22].

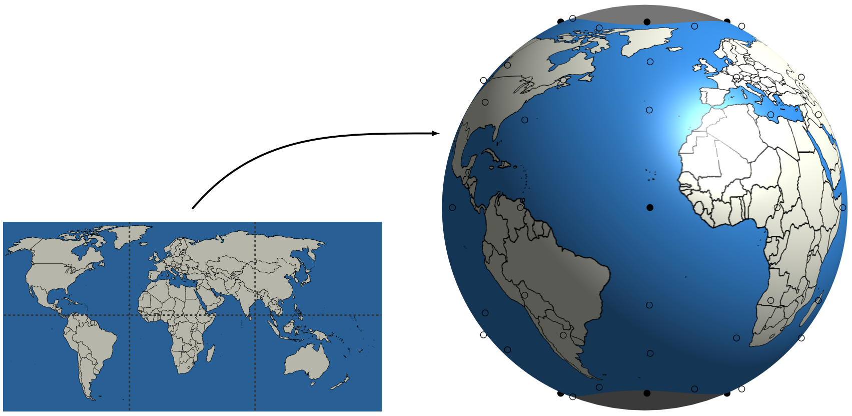


Figure 18: A rectangular map of the world is smoothly mapped onto the sphere via a Bézier spline surface, only fixing a few interpolation points.

6.2 The special orthogonal group SO(d)

Table 2 summarizes the analytic formulas for the necessary Riemannian operators. Figure 19 displays a cubic Bézier spline surface in SO(3) interpolating a random set of interpolation points (red) based on the methods presented in Sections 3.4 and 4 (the optimized control points are shown in green). Note that the spline surface is smooth and roughly follows the control points, but it does not go through them, as expected.

A second example is given in Figure 20, which shows a closed Bézier surface of genus one. Here, the first line is composed of a SO(3) point rotated of $t_1 \times 90$ degrees around the z-axis. In the direction t_2, the data points of the first line are rotated of $t_2 \times 90$ degrees around the x-axis, which gives a torus effect to the figure. The control points have also been optimized, but the method from Section 4 was slightly adapted to account for the periodic boundary: we imposed $u^{0,n}_{k,l} = u^{M,n}_{k,l}$ and $u^{m,0}_{k,l} = u^{m,N}_{k,l}$ for $k, l \in \{-1, 0, 1\}$ and $(m, n) \in \{0, \dots, M\} \times \{0, \dots, N\}$.

Set	$\mathrm{SO}(d) = \{X \in \mathbb{R}^{d\times d} \;:\; X^\top X = I,\ \det(X) = 1\}$
Tangent space	$T_X\mathrm{SO}(n) = \{H \in \mathbb{R}^{d\times d} \;:\; X^\top H + H^\top X = 0\}$
Inner product	$\langle H_1, H_2\rangle = \mathrm{trace}\left(H_1^\top H_2\right)$
Distance	$\mathrm{d}(X, Y) = \lVert \log\left(X^\top Y\right) \rVert_{\mathrm{F}}$
Exponential	$\exp_X(H) = X \exp\left(X^\top H\right)$
Logarithm	$\log_X(Y) = X \log\left(X^\top Y\right)$
Parallel transport	$P_{X\to Y}(H) = Y X^\top H$

Table 2: Riemannian operators for SO(d), *cf.* [5].

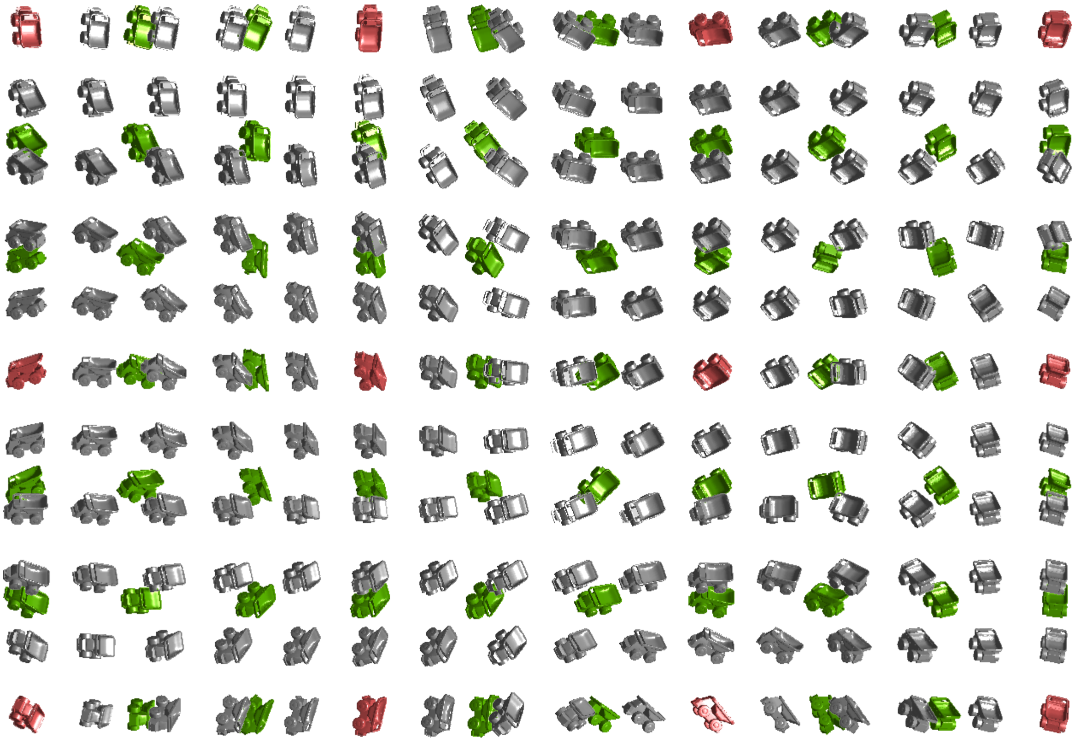


Figure 19: Cubic Bézier spline surface in SO(3) visualized as rotations of an object. Interpolation points in red, optimized control points in green, points on the surface in gray.

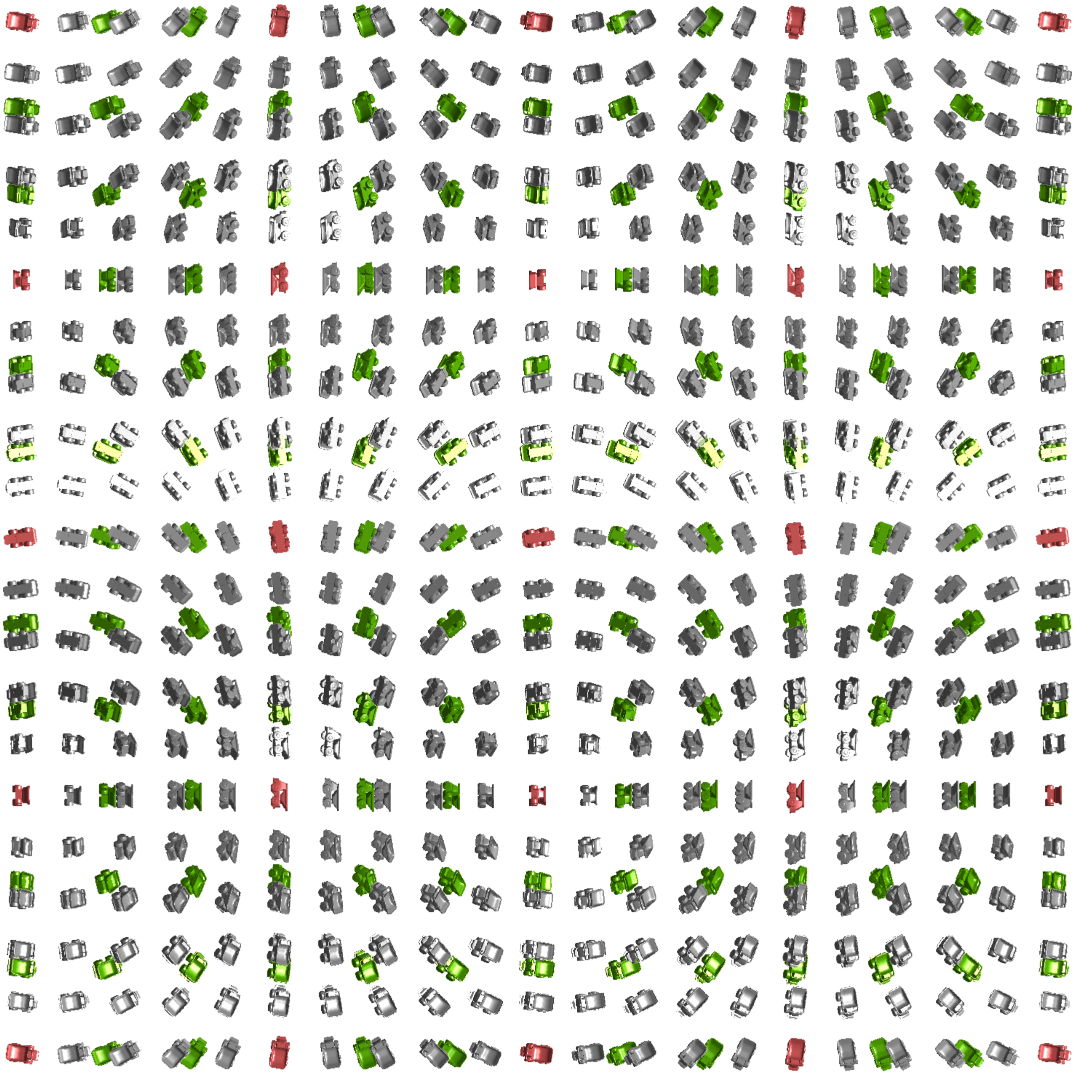


Figure 20: Smooth torus in SO(3) given by a cubic Bézier spline surface (visualization as in Figure 19).

6.3 The space of open polygonal curves $\mathcal{P}$

We consider two shape spaces as further examples. The first is the shape space $\mathcal{P}$ of all polygonal curves in the plane with a fixed number n of segments. Here, two shapes are considered equal if they differ only by a rigid motion. A shape $\gamma \in \mathcal{P}$ can therefore be identified with its segment length and angle representation $(\ell_1, \ldots, \ell_n, \alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{R}^{2n-1}$, where ℓ_j denotes the length of the j^{th} polygon segment and α_j the angle between segment j and $j+1$. The tangent space $T_\gamma \mathcal{P} = \mathbb{R}^{2n-1}$ can be seen as the space of all length and angle variations.

Instead of a Riemannian metric and with regard to Section 5.2, we directly specify an energy functional W acting on two shapes $\gamma_1, \gamma_2 \in \mathcal{P}$, $\gamma_i = (\ell_1^i, \ldots, \ell_n^i, \alpha_1^i, \ldots, \alpha_{n-1}^i)$, $i = 1, 2$, by

$$W[\gamma_1, \gamma_2] = \sum_{j=1}^{n} \frac{(\ell_j^1 - \ell_j^2)^2}{\ell_j^1} + 2 \sum_{j=1}^{n-1} \frac{(\alpha_j^1 - \alpha_j^2)^2}{\ell_j^1 + \ell_{j+1}^1}.$$

The Riemannian metric, for which this W is supposed to approximate the squared Riemannian distance, can be obtained as the second derivative,

$$g_y(v,w) = \frac{\mathrm{d}}{\mathrm{d}t_1}\frac{\mathrm{d}}{\mathrm{d}t_2} W(y, y + t_1 v + t_2 w)\Big|_{t_1=t_2=0},$$

and it has a physical interpretation of energy dissipation during shape deformation [23]. Our calculations are now based on the discrete approximations from Section 5.2.

Figure 21 shows a Bézier spline surface in the space of polygonal curves with optimized control points. The interplation points are segments of silhouettes from the Kimia database [27].

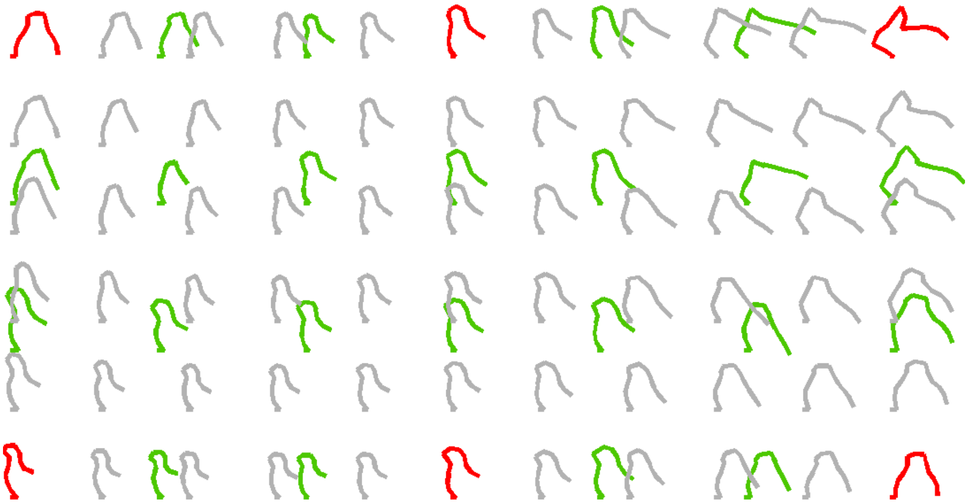


Figure 21: Smooth interpolation on the space of open polygonal curves $\mathcal{P}$. Interpolation points in red, optimized control points in green, points on the surface in gray (shapes are from the Kimia database [27]).

6.4 The space of discrete shells $\mathcal{S}_h$

As a second example, we consider the space of shells $\mathcal{S}_h$, as described in [12]. In the continuous case, a shell S_h is given by an oriented $\mathcal{C}^2$ surface S in $\mathbb{R}^3$, called the *midplane* of S_h, and is defined as the set

$$\mathrm{S}_h = \{p + \lambda\nu(p) \mid p \in \mathrm{S}, \lambda \in \left(-\tfrac{h}{2}, \tfrac{h}{2}\right) \subset \mathbb{R}\}$$

where $\nu(p)$ denotes the normal at a point $p \in \mathrm{S}$. The space of shells $\mathcal{S}_h$ comprises all images $\phi(\mathrm{S}_h^{\mathrm{ref}})$ of a reference shell $\mathrm{S}_h^{\mathrm{ref}}$ under orientation preserving diffeomorphisms ϕ. The tangent space at $\mathrm{S} \in \mathcal{S}$ consists of smooth displacement fields $\psi : \mathrm{S} \to \mathbb{R}^3$, and it can be equipped with a Riemannian metric that describes the physical energy dissipation during the deformation of S_h; for details we refer to [12].

The discretized analog, also described in [12], is a *discrete shell* M_h, by which we mean a triangulated surface in $\mathbb{R}^3$, represented by a tuple $(N_h, T_h) \in (\mathbb{R}^3)^m \times (\{1,\ldots,m\}^3)^n$, $m, n \in \mathbb{N}$. Here, N_h represents the vertex positions and T_h encodes the triangulation (each component $(T_h)_l = (i,j,k)$ indicates that $(N_h)_i$, $(N_h)_j$ and $(N_h)_k$ form a triangle). The space of discrete shells can be equipped with a discrete analog of the Riemannian metric on $\mathcal{S}$. Given a triangle T, we assign to each vertex a local index ranging from 0 to 2. This allows us to define the edge set E_T of T as the set of directed edges connecting the nodes $i-1$ and i (counted modulo 3). E_h is defined to be the union of the edge sets E_T over all triangles T.

A deformation of a discrete shell M_h can now be viewed as a mapping $\phi : N_h \to \mathbb{R}^3$. We define the discrete deformation energy $\tilde{W}[\phi]$ of ϕ by

$$\tilde{W}[\phi] = h \sum_{T \in T_h} W_{\text{mem}}(Q^T_{\text{mem}}[\phi]) A_T + h^3 \sum_{E \in E_h} W_{\text{bend}}(Q^E_{\text{bend}}[\phi]) A_E$$

for some physical energy densities W_{mem} and W_{bend} (specific examples are given in [12]). Here, A_T denotes the area of the (undeformed) triangle T, while for a given edge $E \in E_h$, $A_E = \frac{1}{3}(A_{T_1} + A_{T_2})$ denotes an area fraction of the two adjacent triangles T_1 and T_2. The operators $Q^T_{\text{mem}}[\phi]$ and $Q^E_{\text{bend}}[\phi]$ describe in-plane strain and bending and are given by

$$\begin{aligned} Q^T_{\text{mem}}[\phi] &= B^{\text{mem}}_{\phi,T} - B^{\text{mem}}_{\text{id},T}, \qquad B^{\text{mem}}_{\phi,T} = \frac{1}{8A_T^2} \sum_{i=0}^{2} (I^\phi_j + I^\phi_k + I^\phi_i) e_i \otimes e_i, \qquad {\scriptstyle j=i+1 (\text{mod } 3), \atop k=i+2 (\text{mod } 3),} \\ Q^E_{\text{bend}}[\phi] &= B^{\text{bend}}_{\phi,E} - B^{\text{bend}}_{\text{id},E}, \qquad B^{\text{bend}}_{\phi,E} = \frac{\theta_{\phi(E)}}{A_E} \frac{e \otimes e}{\|E\|}. \end{aligned}$$

Here, e_i denotes the result of a clockwise rotation of E_i by $\pi/2$ in the plane induced by T. Similarly e is the result of a clockwise rotation of E by $\pi/2$ in the plane induced by one of the adjacent triangles. By I^φ_k we denote the squared length of the deformed edge E_k, and $\theta_{\phi(E)}$ stands for the dihedral angle at the deformed edge $\phi(E)$.

The discrete geodesic calculus from Section 5.2 can now be employed with the energy

$$W[\mathrm{S}_1, \mathrm{S}_2] = \tilde{W}[\phi] \qquad \text{for that } \phi \text{ with } \phi(\mathrm{S}_1) = \mathrm{S}_2,$$

which approximates the squared Riemannian distance in the space of discrete shells [12].

Figure 1 already showed a differentiable piecewise-Bézier surface interpolating between six given hand shapes (mesh data made available by Yeh *et al.* [32]). Similarly, Figure 22 shows a piecewise-Bézier interpolation surface between 3×3 interpolation points (the interpolation points in this figure are meshes made available by Bergou *et al.* [4]). The control points in Figure 22 have been optimized using the algorithm from Section 4.

7 Conclusions

We have proposed different generalizations of piecewise-Bézier surfaces in Euclidean space to Riemannian manifolds, all based on geodesic averaging. For those generalizations it is non-trivial to ensure differentiability across the different Bézier patches. We have achieved differentiability by interpreting control points on the boundary of Bézier patches as weighted averages of interior control points. We have furthermore put forward an approach to optimize the control point positions in the Riemannian manifold as to obtain a Bézier surface with small second derivative. Future work might address methods to achieve higher order smoothness of Bézier splines (in principle, similar concepts as in the current work can be expected to apply) or alternative definitions of smooth curves and surfaces in Riemannian manifolds such as curves and surfaces with minimal curvature.

Notations

$\mathcal{M}$	:	Manifold.
$\mathbb{R}^d$	:	Euclidean space of dimension d.
$\mathcal{C}^k$	:	Continuous derivability up to derivative k.

γ	:	Curve on a manifold $\mathcal{M}$.
$L[\gamma]$	:	Length of a curve on a manifold $\mathcal{M}$.
$E[\gamma]$	:	Energy of a curve on a manifold $\mathcal{M}$.
$T_x\mathcal{M}$	:	Tangent space at a point $x \in \mathcal{M}$.
$\exp_a(b)$	:	Exponential map. $a \in \mathcal{M},\ b \in T_a\mathcal{M}$.
$\log_a(b)$	:	Inverse exponential map. $a, b \in \mathcal{M}$.
$P_{a\to b}v$	:	Parallel transport of a vector v from $T_a\mathcal{M}$ to $T_b\mathcal{M}$. $a, b \in \mathcal{M}$.
$\mathrm{EXP}_a(b)$	:	Discrete exponential map. $a \in \mathcal{M},\ b \in T_a\mathcal{M}$.
$\mathrm{LOG}_a(b)$	:	Discrete inverse exponential map. $a, b \in \mathcal{M}$.
$\langle a, b\rangle$	:	Scalar product between $a, b \in T_x\mathcal{M}$.
$\mathrm{d}(a, b)$	:	Distance between $a, b \in \mathcal{M}$.
b	:	Control point of a Bézier function. For curves, it is indexed as b_j and for surfaces b_{ij}. b_{ij}^{mn} is the ij^{th} control point of the patch m, n.
β_K	:	Bézier function of order K (curve or surface).
β^I	:	Bézier function of type I (average over all control points).
β^{II}	:	Bézier function of type II (average as 1D curves in one direction, and then in another).
β^{III}	:	Bézier function of type III (de Casteljau's algorithm).
B_{jK}	:	j^{th} Berstein polynomial of order K.
$\mathfrak{b}$	:	Intermediate point in de Casteljau's algorithm.
$\mathfrak{B}$	:	Bézier spline composed of several Bézier functions.
w	:	Weight.
$\mathrm{av}[(x_1, \ldots, x_n), (w_1, \ldots, w_n)]$	:	Geodesic average of points $x_i \in \mathcal{M}$ with weights $w_i \in \mathbb{R}$.
$v_{i,j}^{m,n}(r, s)$	:	Mapping of a point i, j in the patch m, n to the tangent space of the control point $b_{0,0}^{m+r,n+s}$.

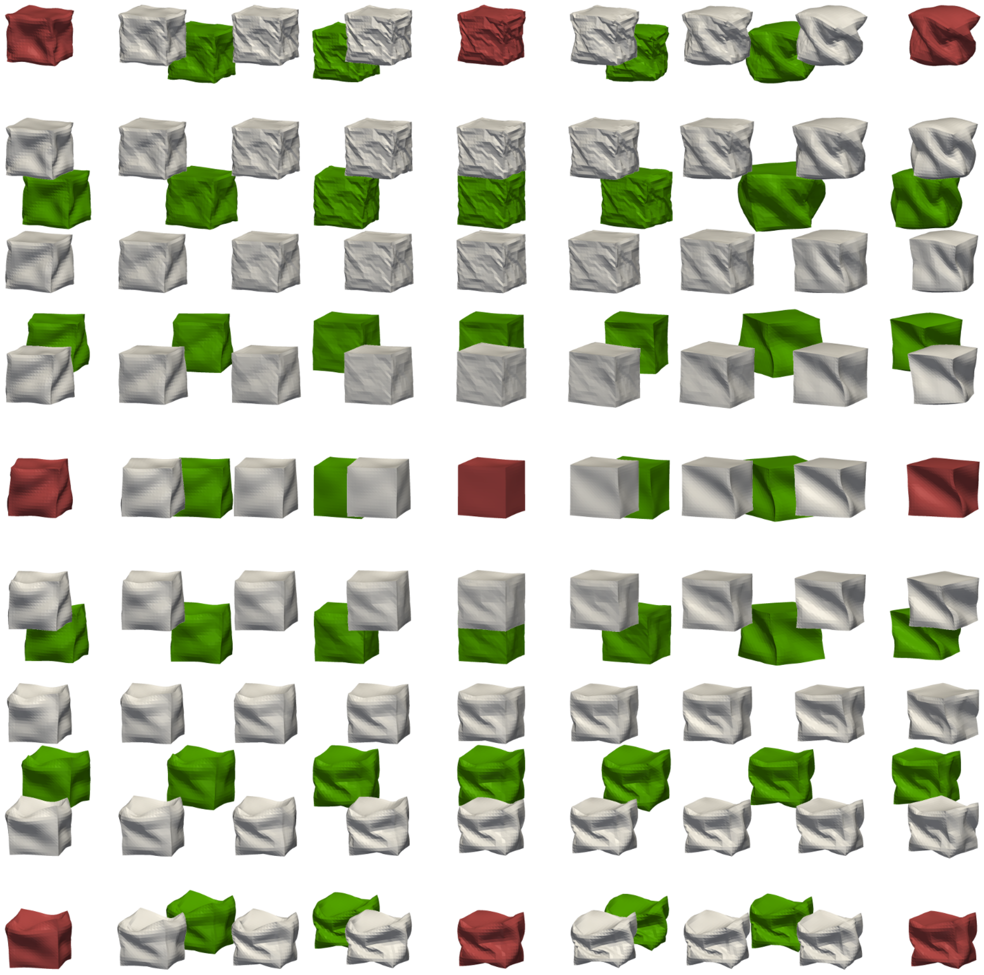


Figure 22: Differentiable piecewise-Bézier interpolation in the space of triangulated shells. Interpolation points in red, optimized control points in green, points on the surface in gray.

Acknowledgements

The authors thank Behrend Heeren and Martin Rumpf for providing the code framework for calculations in shell space. B.W.'s research was supported by the Alfried Krupp Prize for Young University Teachers awarded by the Alfried Krupp von Bohlen und Halbach-Stiftung. The work was also supported by the Deutsche Forschungsgemeinschaft (DFG), Cells-in-Motion Cluster of Excellence (EXC1003 – CiM), University of Münster, Germany. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. This work was supported by the Belgian FNRS through grant FRFC 2.4585.12.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[2] A Arnould, P.-Y. Gousenbourger, C Samir, P.-A. Absil, and M Canis. Fitting Smooth Paths on Riemannian Manifolds : Endometrial Surface Reconstruction and Preoperative MRI-Based Navigation. In F.Nielsen and F.Barbaresco, editors, *GSI2015*, pages 491–498. Springer International Publishing, 2015.

[3] Ronny Bergmann, Friederike Laus, Gabriele Steidl, and Andreas Weinmann. Second order differences of cyclic data and applications in variational denoising. *SIAM J. Imaging Sci.*, 7(4):2916–2953, 2014.

[4] Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun. TRACKS: Toward Directable Thin Shells. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):50:1–50:10, jul 2007.

[5] Nicolas Boumal and P.-A. Absil. A discrete regression method on manifolds and its application to data on SO(n). In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 18, pages 2284–2289, 2011.

[6] Nicolas Boumal, Bamdev Mishra, P.-A. Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[7] M.P. do Carmo. *Riemannian Geometry*. Mathematics (Birkhäuser) theory. Birkhäuser Boston, 1992.

[8] G. Farin. *Curves and Surfaces for CAGD*. Academic Press, fifth edition, 2002.

[9] P. Thomas Fletcher. Geodesic regression and the theory of least squares on Riemannian manifolds. *Int. J. Comput. Vis.*, 105(2):171–185, 2013.

[10] M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Ann. Inst. H. Poincaré*, 10:215–310, 1948.

[11] Pierre-Yves Gousenbourger, Chafik Samir, and P.-A. Absil. Piecewise-Bézier C1 interpolation on Riemannian manifolds with application to 2D shape morphing. In *International Conference on Pattern Recognition*, 2014.

[12] B. Heeren, M. Rumpf, M. Wardetzky, and B. Wirth. Time-discrete geodesics in the space of shells. *Computer Graphics Forum*, 31(5):1755–1764, 2012.

[13] Jacob Hinkle, Prasanna Muralidharan, P.Thomas Fletcher, and Sarang Joshi. Polynomial regression on Riemannian manifolds. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision ECCV 2012*, volume 7574 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2012.

[14] A.A. Joshi, D.W. Shattuck, P.M. Thompson, and R.M. Leahy. Surface-constrained volumetric brain registration using harmonic mappings. *Medical Imaging, IEEE Transactions on*, 26(12):1657–1669, Dec 2007.

[15] Jürgen Jost. *Nonpositive curvature: geometric and analytic aspects*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 1997.

[16] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5):509–541, 1977.

[17] A. Lin and M. Walker. Cgd techniques for differentiable manifolds. *Proceedings of the 2001 International Symposium Algorithms for Approximation IV*, 2001.

[18] Domenico Mucci. Maps into projective spaces: liquid crystal and conformal energies. *Discrete & Continuous Dynamical Systems - Series B*, 17(2):597–635, 2012.

[19] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[20] T. Popiel and L. Noakes. Bézier curves and $\mathcal{C}^2$ interpolation in Riemannian manifolds. *J. Approx. Theory*, pages 148(2):111–127, 2007.

[21] Lorenz Pyta and Dirk Abel. Model based control of the incompressible Navier-Stokes-equations using interpolatory model reduction, 2015. To appear in the proceedings of the 54th IEEE Conference on Decision and Control.

[22] Q. Rentmeesters. A gradient method for geodesic data fitting on some symmetric Riemannian manifolds. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 7141–7146, 2011.

[23] Martin Rumpf and Benedikt Wirth. Variational time discretization of geodesic calculus. *IMA Journal of Numerical Analysis*, 2014. doi:10.1093/imanum/dru027.

[24] C. Samir, P.-A. Absil, A. Srivastava, and E. Klassen. A gradient-descent method for curve fitting on Riemannian manifolds. *Foundations of Computational Mathematics*, 12:49–73, 2012.

[25] Oliver Sander. Geodesic finite elements for Cosserat rods. *International Journal for Numerical Methods in Engineering*, 82(13):1645–1670, 2010.

[26] Oliver Sander, Patrizio Neff, and Mircea Brsan. Numerical treatment of a geometrically nonlinear planar Cosserat shell model, 2015.

[27] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing their shock graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):550–571, 2004.

[28] F. Steinke, M. Hein, J. Peters, and B. Schölkopf. Manifold-valued thin-plate splines with applications in computer graphics. *Computer Graphics Forum*, 27(2):437–448, apr 2008.

[29] Florian Steinke, Matthias Hein, and Bernhard Schölkopf. Nonparametric regression between general Riemannian manifolds. *SIAM Journal on Imaging Sciences*, 3(3):527–563, 2010.

[30] J. Su, I.L. Dryden, E. Klassen, H. Le, and A. Srivastava. Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds. *Image and Vision Computing*, 30(67):428 – 442, 2012.

[31] Bart Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013.

[32] I-Cheng Yeh, Chao-Hung Lin, Olga Sorkine, and Tong-Yee Lee. Template-based 3d model fitting using dual-domain relaxation. *IEEE Transactions on Visualization and Computer Graphics*, 99(RapidPosts), 2010.