frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 12 December 2018
doi: 10.3389/fams.2018.00059

A Variational Model for Data Fitting on Manifolds by Minimizing the Acceleration of a Bézier Curve

Ronny Bergmann[1]* *and Pierre-Yves Gousenbourger*[2]

[1] Research Group Numerical Mathematics (Partial Differential Equations), Faculty of Mathematics, Technische Universität Chemnitz, Chemnitz, Germany, [2] ICTEAM Institute, Université Catholique de Louvain, Louvain-la-Neuve, Belgium

We derive a variational model to fit a composite Bézier curve to a set of data points on a Riemannian manifold. The resulting curve is obtained in such a way that its mean squared acceleration is minimal in addition to remaining close the data points. We approximate the acceleration by discretizing the squared second order derivative along the curve. We derive a closed-form, numerically stable and efficient algorithm to compute the gradient of a Bézier curve on manifolds with respect to its control points, expressed as a concatenation of so-called adjoint Jacobi fields. Several examples illustrate the capabilities and validity of this approach both for interpolation and approximation. The examples also illustrate that the approach outperforms previous works tackling this problem.

AMS subject classification (2010). 65K10, 65D10, 65D25, 53C22, 49Q99.

Keywords: Riemannian manifolds, curve fitting, composite Bézier curves, Jacobi fields, variational models

OPEN ACCESS

Edited by:
Luca Marchetti, Microsoft Research-University of Trento Centre for Computational and Systems Biology(COSBI), Italy

Reviewed by:
Giovanni Mascali, Università della Calabria, Italy
Vincenzo Bonnici, Università degli Studi di Verona, Italy

***Correspondence:**
Ronny Bergmann
ronny.bergmann@mathematik.tu-chemnitz.de

Specialty section:
This article was submitted to Optimization, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *26 July 2018*
Accepted: *13 November 2018*
Published: *12 December 2018*

Citation:
Bergmann R and Gousenbourger P-Y (2018) A Variational Model for Data Fitting on Manifolds by Minimizing the Acceleration of a Bézier Curve. Front. Appl. Math. Stat. 4:59. doi: 10.3389/fams.2018.00059

1. INTRODUCTION

This papers addresses the problem of fitting a smooth curve to data points $d_0, \dots, d_n$ lying on a Riemannian manifold $\mathcal{M}$ and associated with real-valued parameters $t_0, \dots, t_n$. The curve strikes a balance between a data proximity constraint and a smoothing regularization constraint.

Several applications motivate this problem in engineering and the sciences. For instance, curve fitting is of high interest in projection-based model order reduction of one-dimensional dynamical systems [1]. In that application, the dynamical system depends on the Reynolds number and the model reduction is obtained by computing suitable projectors as points on a Grassmann manifold. Finding a projector is however a time- and memory-consuming task. Based on projectors precomputed for given parameter values, fitting is used to approximate the projector associated with a new parameter value. In Gousenbourger et al. [2], the same strategy is used to approximate wind field orientations represented as points on the manifold of positive semidefinite covariance matrices of size p and rank r. Further applications are involving rigid-body motions on SE(3), like Cosserat rods applications [3], or orientation tracking [4]. Sphere-valued data are also of interest in many applications of data analysis, for storm tracking and prediction, or the study of bird migration [5].

There exists different approaches to tackle the curve fitting problem. Among others, we name here the subdivision schemes approach [6, 7] or the Lie-algebraic methods [8]. However, the most

popular approach nowadays is probably to encapsulate the two above-mentioned constraints into an optimization problem

$$\min_{\gamma\in\Gamma} E_\lambda(\gamma) \coloneqq \min_{\gamma\in\Gamma} \int_{t_0}^{t_n} \left\| \frac{\mathrm{D}^2\gamma(t)}{\mathrm{d}t^2} \right\|^2_{\gamma(t)} \mathrm{d}t + \frac{\lambda}{2}\sum_{i=0}^{n} \mathrm{d}^2\big(\gamma(t_i), d_i\big), \tag{1}$$

where Γ is an admissible space of curves $\gamma\colon [t_0, t_n] \to \mathcal{M}$, $t \mapsto \gamma(t)$, $\frac{\mathrm{D}^2}{\mathrm{d}t^2}$ denotes the (Levi-Civita) second covariant derivative, and $\|\cdot\|_{\gamma(t)}$ denotes the Riemannian metric at $\gamma(t)$ from which we can define a Riemannian distance $\mathrm{d}(\cdot,\cdot)$. Finally, $\lambda \in \mathbb{R}$ is a parameter that strikes the balance between the *regularizer* $\int_{t_0}^{t_n} \left\| \frac{\mathrm{D}^2\gamma(t)}{\mathrm{d}t^2} \right\|^2_{\gamma(t)} \mathrm{d}t$ and the *fitting term* $\sum_{i=0}^{n} \mathrm{d}^2(\gamma(t_i), d_i))$. This approach leads to the remarkable property that, when the manifold $\mathcal{M}$ reduces to the Euclidean space and Γ is the Sobolev space $H^2(t_0, t_n)$, the solution to (1) is the natural cubic spline [9].

Optimization on manifolds has gained a lot of interest this last decade, starting with the textbook [10] that summarizes several optimization methods on matrix manifolds. Recently, toolboxes have emerged, providing easy access to such optimization methods, e.g., Manopt [11] and MVIRT [12]. The former received a very positive return in many different topics of research, with applications for example in low-rank modeling in image analysis [13], dimensionality reduction [14], phase retrieval [15] or even 5G-like MIMO systems [16]. The latter stems from recent interest in manifold-valued image and data processing, phrased as variational models on the product manifold $\mathcal{M}^N$, where N is the number of pixels. Starting with total variation (TV) regularization of phase-valued data [17, 18], different methods for TV on manifolds [19, 20] have been developed as well as second order methods [21, 22] up to infimal convolution [23] and total generalized variation [23, 24]. Furthermore, different algorithms have been generalized to manifold-valued data, besides the previous works using gradient descent or cyclic proximal point methods, a Douglas–Rachford splitting [25], iteratively reweighted least squares [26] and more general half-quadratic minimization [27] have been introduced.

The curve fitting problem (1), has been tackled differently the past few years. Samir et al. [28] considered the case where Γ is an infinite dimensional Sobolev space of curves, and used the Palais-metric to design a gradient descent algorithm for (1) (see e.g., [29] for an application of this approach). Another method consists in discretizing the curve γ in N points, and therefore considering $\Gamma = \mathcal{M}^N$ (see e.g., [30] for a result on SO(3)). Finally, the limit case where $\lambda \to 0$ is already well studied and known as the geodesic regression [31–33].

A recent topic concerns curve fitting by means of Bézier curves. In that approach, the search space Γ is reduced to as set of composite Bézier curves. Those are a very versatile tool to model smooth curves and surfaces for real- and vector-valued discrete data points (see [34] for a comprehensive textbook), but they can also be used to model smooth curves and surfaces for manifold-valued data [35, 36]. The advantage to work with such objects, compared to classical approaches, are that (i) the search space is drastically reduced to the so-called control points of the Bézier curves (and this leads to better time and memory performances) and (ii) it is very simple to impose differentiability for the optimal curve, which is appreciated in several of the above-mentioned applications. However, while obtaining such an optimal curve reduces directly to solving a linear system of equations for data given on a Euclidean space, there is up to now no known closed form of the optimal Bézier curve for manifold valued data.

In this work, we derive a gradient descent algorithm to compute a differentiable composite Bézier curve $\mathbf{B}\colon [t_0, t_n] \to \mathcal{M}$ that satisfies (1), i.e., such that $\mathbf{B}(t)$ has a minimal mean squared acceleration, and fits the set of $n+1$ manifold-valued data points at their associated time-parameters. We consider the manifold-valued generalization of Bézier curves [36] in the same setting as in Arnould et al. [37], or more recently in Gousenbourger et al. [38]. We employ the (squared) second order absolute difference introduced in Bačák et al. [22] to obtain a discrete approximation of the regularizer from (1). The quality of the approximation depends only on the number of sampling points. We exploit the recursive structure of the De Casteljau algorithm [36] to derive the gradient of the objective function with respect to the control points of $\mathbf{B}(t)$. The gradient is built as a recursion of Jacobi fields that, for numerical reasons, are implemented as a concatenation of so-called *adjoint Jacobi fields*. Furthermore, the corresponding variational model only depends on the number of control points of the composite Bézier curve, and not on the number of sampling points. We finally obtain an approximating model to (1) that we solve with an algorithm only based on three tools on the manifold: the exponential map, the logarithmic map, and a certain Jacobi field along geodesics.

The paper is organized as follows. We introduce the necessary preliminaries—Bézier curves, Riemannian manifolds and Riemannian second order finite differences— in section 2. In section 3 we derive the gradient of the discretized mean squared acceleration of the composite Bézier curve with respect to its control points, and thus of the regularizer of (1). In section 4, we present the corresponding gradient descent algorithm, as well as an efficient gradient evaluation method, to solve (1) for different values of λ. The limit case where $\lambda \to \infty$ is studied as well. Finally, in section 5, we validate, analyze and illustrate the performance of the algorithm for several numerical examples on the sphere $\mathbb{S}^2$ and on the special orthogonal group SO(3). We also compare our solution to existing Bézier fitting methods. A conclusion is given in section 6.

2. PRELIMINARIES

2.1. Bézier Functions and Composite Bézier Spline

Consider the Euclidean space $\mathbb{R}^m$. A *Bézier curve of degree* $K \in \mathbb{N}$ is a function $\beta_K\colon [0,1] \to \mathbb{R}^m$ parametrized by *control points* $b_0, \ldots, b_K \in \mathbb{R}^m$ and taking the form

$$\beta_K(t; b_0, \ldots, b_K) \coloneqq \sum_{j=0}^{K} b_j B_{j,K}(t),$$

where $B_{j,K}(t) = \binom{K}{j} t^j (1-t)^{K-j}$ are the *Bernstein basis polynomials* [34]. For example the linear Bézier curve β_1 is just

the line segment $(1-t)b_0+tb_1$ connecting the two control points b_0 and b_1. The explicit formulae of the quadratic and cubic Bézier curves read

$$\beta_2(t;b_0,b_1,b_2) = b_0(1-t)^2 + 2b_1(1-t)t + b_2t^2,$$
$$\beta_3(t;b_0,b_1,b_2,b_3) = b_0(1-t)^3 + 3b_1(1-t)^2t + 3b_2(1-t)t^2 + b_3t^3,$$

for given control points $b_0, b_1, b_2 \in \mathbb{R}^m$ and an additional point $b_3 \in \mathbb{R}^m$ for the cubic case.

A *composite Bézier curve* is a function $\mathbf{B}\colon [0,n] \to \mathbb{R}^m$ composed of n Bézier curves and defined as

$$\mathbf{B}(t) := \begin{cases} \beta_{K_0}(t; b_0^0, \dots, b_{K_0}^0) & \text{if } t \in [0,1] \\ \beta_{K_i}(t-i; b_0^i, \dots, b_{K_i}^i) & \text{if } t \in (i, i+1], \\ & i \in \{1, \dots, n-1\}, \end{cases} \tag{2}$$

where $K_i \in \mathbb{N}$ denotes the degree of the ith Bézier curve β_{K_i} of $\mathbf{B}$ and b_j^i, $j = 0, \dots, K_i$, are its control points. Furthermore, $\mathbf{B}(t)$ is continuous if the last and first control points of two consecutive segments coincide [34]. We introduce p_i as the point at the junction of two consecutive Bézier segments, i.e., $p_i := b_{K_{i-1}}^{i-1} = b_0^i$. Differentiability is obtained if the control points of two consecutive segments are aligned. We introduce further $b_{i+1}^- := b_{K_i-1}^i$ and $b_{i+1}^+ := b_1^{i+1}$ such that the differentiability condition reads $p_{i+1} = \frac{K_i b_{i+1}^- + K_{i+1} b_+^{i+1}}{K_i + K_{i+1}}$.

Example 1 (C^1 conditions for composite cubic Bézier curves)**.** *We consider the case where the Bézier segments are all cubic, as represented in* ***Figure 1****. The composite Bézier curve* $\mathbf{B}(t)$ *is* C^1 *if* $p_i = \frac{1}{2}(b_i^- + b_i^+)$, $i = 1, \dots, 4$, *with* $p_i = b_3^{i-1} = b_0^i$, $b_i^- = b_2^{i-1}$, *and* $b_i^+ = b_1^i$.

2.2. Riemannian Manifolds

We consider a complete m-dimensional Riemannian manifold $\mathcal{M}$. We refer to O'Neill [39] and do Carmo [40] for an introduction to Riemannian manifolds and to [10] for optimization thereon. We will use the following terminology and notations.

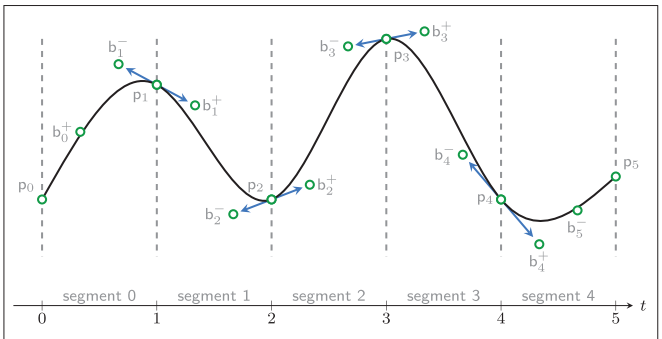

FIGURE 1 | Schematic representation of the composite cubic Bézier curve $\mathbf{B}:[0,5] \to \mathcal{M}$, for $\mathcal{M} = \mathbb{R}$ (black), and its control points (green circles). Continuous differentiability is reached at the junction of the segments (the blue arrows draw the first derivative of $\mathbf{B}$).

We denote by $T_a\mathcal{M}$ the (Euclidean) tangent space to $\mathcal{M}$ at $a \in \mathcal{M}$; $T\mathcal{M} := \cup_{a\in\mathcal{M}} T_a\mathcal{M}$ is the tangent bundle to $\mathcal{M}$; $\langle\cdot,\cdot\rangle_a$ denotes the inner product in the tangent space $T_a\mathcal{M}$ at a and from which we deduce the norm of $v \in T_a\mathcal{M}$ denoted by $\|v\|_a = \sqrt{\langle v, v\rangle_a}$. For a (not necessarily unique) shortest geodesic between a and $b \in \mathcal{M}$, we write $g(\cdot; a, b)\colon \mathbb{R} \to \mathcal{M}$, $t \mapsto g(t; a, b)$, parametrized such that $g(0; a, b) = a$ and $g(1; a, b) = b$. This choice of parametrization also means that the covariant derivative $\frac{D}{dt}$ of g with respect to time satisfies $\|\frac{D}{dt}g(t; a, b)\|_{g(t)} = d_{\mathcal{M}}(a, b)$, for all $t \in [0, 1]$, where $d_{\mathcal{M}}(a, b)$ is the geodesic distance between a and $b \in \mathcal{M}$. The Riemannian exponential reads $\exp_a\colon T_a\mathcal{M} \to \mathcal{M}, v \mapsto b = \exp_a(v)$ and we denote by $r_a \in \mathbb{R}$ the maximal radius such that the exponential map is bijective on $\mathcal{D}_a := \{b \in \mathcal{M} : d_{\mathcal{M}}(a, b) < r_a\}$. Then $\log_a\colon \mathcal{D}_a \to T_a\mathcal{M}, b \mapsto v = \log_a(b)$ is called the Riemannian logarithm which is (locally) the inverse of the exponential. A Riemannian manifold is called symmetric in $x \in \mathcal{M}$ if the geodesic reflection s_x at $x \in \mathcal{M}$ given by the mapping $\gamma(t) \mapsto \gamma(-t)$ is an isometry at least locally near x, for all geodesics through $\gamma(0) = x$. If $\mathcal{M}$ is symmetric in every $x \in \mathcal{M}$, the manifold is called (Riemannian) symmetric space or symmetric manifold.

In the following we assume, that both the exponential and the logarithmic map are available for the manifold and that they are computationally not too expensive to evaluate. Furthermore, we assume that the manifold is symmetric.

2.3. Composite Bézier Curves on Manifolds

One well-known way to generalize Bézier curves to a Riemannian manifold $\mathcal{M}$ is via the De Casteljau algorithm [36, section 2]. This algorithm only requires the Riemannian exponential and logarithm and conserves the interpolation property of the first and last control points. Some examples on interpolation and fitting with Bézier curves or Bézier surfaces, i.e., generalizations of tensor product Bézier curves, can be found in Absil et al. [35].

Consider $\beta_K\colon [0,1] \to \mathcal{M}, t \mapsto \beta_K(t; b_0, b_1, \dots, b_K)$, the manifold-valued Bézier curve of order K driven by $K+1$ control points $b_0, \dots, b_K \in \mathcal{M}$. We introduce the points $x_i^{[0]} = b_i$ and iterate the construction of further points. For $i = 0, \dots, K-k$, $k = 1, \dots, K$, we define

$$x_i^{[k]} := \beta_k(t; b_i, \dots, b_{i+k}) = g(t; x_i^{[k-1]}, x_{i+1}^{[k-1]}) \tag{3}$$

as the ith point of the kth step of the De Casteljau algorithm, and obtain $\beta_K(t; b_0, \dots, b_K) = x_0^{[K]}$.

The De Casteljau algorithm is illustrated on **Figure 2** for a Euclidean cubic Bézier curve $\beta_3(t; b_0, b_1, b_2, b_3)$. The general cubic Bézier curve can be explicitly expressed on a manifold $\mathcal{M}$ as

$$\begin{aligned}\beta_3(t; x_0^{[0]}, x_1^{[0]}, x_2^{[0]}, x_3^{[0]}) &= g\Big(t; g\big(t; g(t; x_0^{[0]}, x_1^{[0]}), g(t; x_1^{[0]}, x_2^{[0]})\big), \\ &\qquad g\big(t; g(t; x_1^{[0]}, x_2^{[0]}), g(t; x_2^{[0]}, x_3^{[0]})\big)\Big) \\ &= g\big(t; g(t; x_0^{[1]}, x_1^{[1]}), g(t; x_1^{[1]}, x_2^{[1]})\big) \\ &= g(t; x_0^{[2]}, x_1^{[2]}) \\ &= x_0^{[3]}.\end{aligned}$$

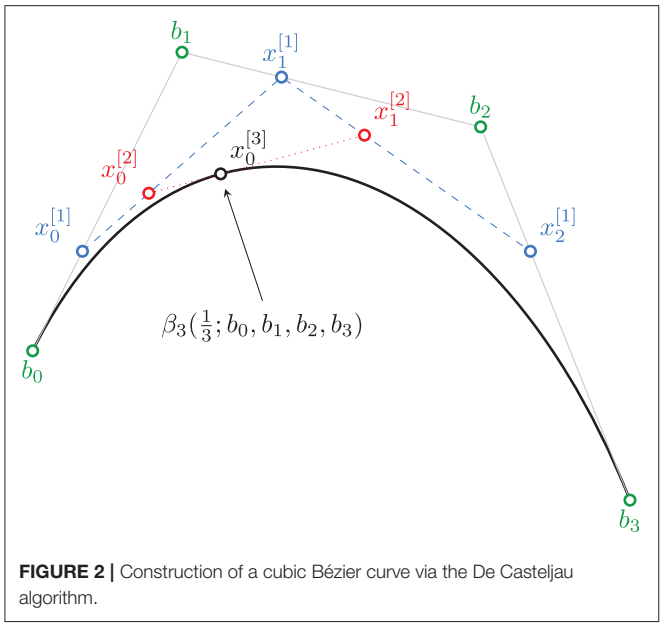

FIGURE 2 | Construction of a cubic Bézier curve via the De Casteljau algorithm.

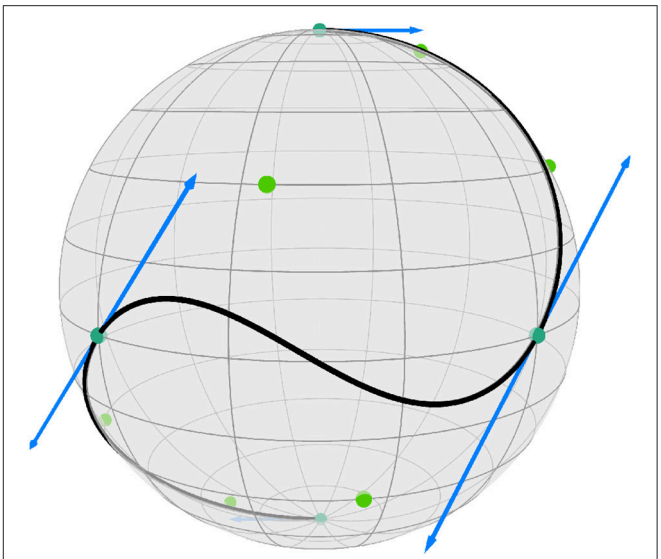

FIGURE 3 | A composite cubic Bézier curve $\mathbf{B}\colon [0,3] \to \mathbb{S}^2$. The end points p_i, $i = 0, \dots, 3$, (cyan) and intermediate points $b_i^{\pm}$ (green) determine its shape; continuous differentiability is illustrated by the logarithmic maps $\log_{p_i} b_i^{\pm}$, $i \in \{1, 2\}$ (blue arrows).

The conditions of continuity and differentiability are generalized to manifolds in Popiel and Noakes [36].

Lemma 2 (Differentiability conditions [36]). *Consider the composite Bézier curve* $\mathbf{B}\colon [0,2] \to \mathcal{M}$ *consisting of* $f\colon [0,1] \to \mathcal{M}, t \mapsto f(t) = \beta_K(t; b_0^0, b_1^0, \dots, b_K^0)$ *and* $g\colon [1,2] \to \mathcal{M}, t \mapsto g(t) = \beta_{\bar{K}}(t-1; b_0^1, b_1^1, \dots, b_{\bar{K}}^1)$, *i.e.,*

$$\mathbf{B}(t) := \begin{cases} f(t) & \text{for } t \in [0,1], \\ g(t) & \text{for } t \in (1,2]. \end{cases}$$

The composite Bézier curve $\mathbf{B}(t)$ *is continuous and continuously differentiable if the two following conditions hold:*

$$b_K^0 = b_0^1 \qquad \text{and} \qquad b_K^0 = g(s; b_{K-1}^0, b_1^1), \quad s = \frac{K}{K+\bar{K}}. \tag{4}$$

The composite Bézier curve $\mathbf{B}\colon [0,n] \to \mathcal{M}$ is then defined completely analogously to the Euclidean case from Equation (2). The differentiability conditions (4) of Lemma 2 have to be satisfied at each junction point p_i, $i = 1, \dots, n-1$.

Example 3 (Composite cubic Bézier curves). *The composite cubic Bézier curve* $\mathbf{B}(t)\colon [0,n] \to \mathcal{M}$ *is* C^1 *if* $p_i = g(\frac{1}{2}; b_i^-, b_i^+)$. *See* ***Figure 1*** *for an example on* $\mathcal{M} = \mathbb{R}$ *with* $n = 5$ *segments and* ***Figure 3*** *for an example on* $\mathcal{M} = \mathbb{S}^2$ *with* $n = 3$ *segments.*

2.4. Discrete Approximation of the Mean Squared Acceleration

We discretize the mean squared acceleration (MSA) of a curve $\gamma\colon [0,1] \to \mathcal{M}$, by discretizing the corresponding integral

$$\int_0^1 \left\| \frac{D^2\gamma(t)}{\mathrm{d}t^2} \right\|_{\gamma(t)}^2 \mathrm{d}t, \tag{5}$$

i.e., the regularizer from (1). We approximate the squared norm of the second (covariant) derivative by the second order absolute finite difference introduced by Bačák et al. [22]. Consider three points $x, y, z \in \mathcal{M}$ and the *set of mid-points of* x *and* z

$$\mathcal{C}_{x,z} := \left\{ c \,\middle|\, c = g(\tfrac{1}{2}; x, z) \right\},$$

for all (not necessarily shortest) geodesics g connecting x and z. The *manifold-valued second order absolute finite difference* is defined by

$$\mathrm{d}_2[x,y,z] = \min_{c \in \mathcal{C}_{x,z}} 2d_{\mathcal{M}}(c,y). \tag{6}$$

This definition is equivalent, on the Euclidean space, to $\|x - 2y + z\| = 2\|\frac{1}{2}(x+z) - y\|$.

Using equispaced points $t_0, \dots, t_N$, $N \in \mathbb{N}$, with step size $\Delta_t = t_1 - t_0 = \frac{1}{N}$, we approximate $\left\| \frac{D^2\gamma(t_i)}{\mathrm{d}t^2} \right\|_{\gamma(t_i)} \approx \frac{1}{\Delta_t^2} d_2[\gamma(t_{i-1}), \gamma(t_i), \gamma(t_{i+1})]$, $i = 1, \dots, N-1$, and obtain by the trapezoidal rule

$$\int_0^1 \left\| \frac{D^2\gamma(t)}{\mathrm{d}t^2} \right\|_{\gamma(t)}^2 \mathrm{d}t \approx \sum_{k=1}^{N-1} \frac{\Delta_t d_2^2[\gamma(t_{i-1}), \gamma(t_i), \gamma(t_{i+1})]}{\Delta_t^4}.$$

For Bézier curves $\gamma(t) = \mathbf{B}(t)$ we obtain for the regularizer in (1) the *discretized MSA* $A(\mathbf{b})$ that depends on the control points b and reads

$$A(\mathbf{b}) := \sum_{i=1}^{N-1} \frac{\mathrm{d}_2^2\left[\mathbf{B}(t_{i-1}), \mathbf{B}(t_i), \mathbf{B}(t_{i+1})\right]}{\Delta_t^3}. \tag{7}$$

3. THE GRADIENT OF THE DISCRETIZED MEAN SQUARED ACCELERATION

In order to minimize the discretized MSA $A(\mathbf{b})$, we aim to employ a gradient descent algorithm on the product manifold $\mathcal{M}^M$, where M is the number of elements in b. In the following, we derive a closed form of the gradient $\nabla_{\mathbf{b}}A(\mathbf{b})$ of the discretized MSA (7). This gradient is obtained by means of a recursion and the chain rule. In fact, the derivative of (6) is already known [22], such that it only remains to compute the derivative of the composite Bézier curve.

We first introduce the following notation. We denote by $D_x f[\eta](x_0) \in T_{f(x_0)}\mathcal{M}$ the directional derivative of $f\colon \mathcal{M} \to \mathcal{M}$ evaluated at x_0, with respect to its argument x and in the direction $\eta \in T_x\mathcal{M}$. We use the short hand $D_x f[\eta] = D_x f[\eta](x)$ whenever this directional derivative is evaluated afterwards again at x.

We now state the two following definitions, which are crucial for the rest of this section.

Definition 4 (Gradient [10, Equation (3.31), p. 46])**.** *Let $f\colon \mathcal{M} \to \mathbb{R}$ be a real-valued function on a manifold $\mathcal{M}$, $x \in \mathcal{M}$ and $\eta \in T_x\mathcal{M}$.*

The gradient $\nabla_{\mathcal{M}} f(x) \in T_x\mathcal{M}$ of f at x is defined as the tangent vector that fulfills

$$\langle \nabla_{\mathcal{M}} f(x), \eta\rangle_x = D_x f[\eta] \quad \text{for all} \quad \eta \in T_x\mathcal{M}. \tag{8}$$

For multivariate functions $f(x, y)$, we denote the gradient of f with respect to x at (x_0, y_0) by writing $\nabla_{\mathcal{M},x} f(x_0, y_0)$. We shorten this notation as $\nabla_{\mathcal{M},x} f = \nabla_{\mathcal{M},x} f(x, y)$ when this gradient is seen as a function of x (and y).

Definition 5 (Chain rule on manifolds [10, p. 195])**.** *Let $f\colon \mathcal{M} \to \mathcal{M}$, $h\colon \mathcal{M} \to \mathcal{M}$ be two functions on a manifold $\mathcal{M}$ and $F\colon \mathcal{M} \to \mathcal{M}$, $x \mapsto F(x) = (f \circ h)(x) = f(h(x))$, their composition. Let $x \in \mathcal{M}$ and $\eta \in T_x\mathcal{M}$. The directional derivative $D_x F[\eta]$ of F with respect to x in the direction η is given by*

$$D_x F[\eta] = D_{h(x)} f\big[D_x h[\eta]\big], \tag{9}$$

where $D_x h[\eta] \in T_{h(x)}\mathcal{M}$ and $D_x F[\eta] \in T_{F(x)}\mathcal{M}$.

The remainder of this section is organized in four parts. We first recall the theory on Jacobi fields in section 3.1 and their relation to the differential of geodesics (with respect to start and end point). In section 3.2, we apply the chain rule to the composition of two geodesics, which appears within the De Casteljau algorithm. We use this result to build an algorithmic derivation of the differential of a general Bézier curve on manifolds with respect to its control points (section 3.3). We extend the result to composite Bézier curves in section 3.4, including their constraints on junction points p_i to enforce the C^1 condition (4), and finally gather these results to state the gradient $\nabla_{\mathcal{M}} A(\mathbf{b})$ of the discretized MSA (7) with respect to the control points.

3.1. Jacobi Fields as Derivative of a Geodesic

In the following, we introduce a closed form of the differential $D_x g(t; \cdot, y)$ of a geodesic $g(t; x, y)$, $t \in [0, 1]$, with respect to its start point $x \in \mathcal{M}$. The differential with respect to the end point $y \in \mathcal{M}$ can be obtained by taking the geodesic $g(t, y, x) = g(1 - t; x, y)$.

As represented in **Figure 4**, we denote by $\gamma_{x,\xi}$, the geodesic starting in $\gamma_{x,\xi}(0) = x$ and with direction $\frac{D}{dt}\gamma_{x,\xi}(0) = \xi \in T_x\mathcal{M}$. We introduce $\zeta(s) := \log_{\gamma_{x,\xi}(s)} y$, the tangential vector in $T_{\gamma_{x,\xi}(s)}\mathcal{M}$ pointing toward y. Then, the *geodesic variation* $\Gamma_{g,\xi}(s, t)$ of the geodesic $g(\cdot; x, y)$ with respect to the tangential direction $\xi \in T_x\mathcal{M}$ is given by

$$\Gamma_{g,\xi}(s, t) := \exp_{\gamma_{x,\xi}(s)}(t\zeta(s)), \qquad s \in (-\varepsilon, \varepsilon),\ t \in [0, 1],$$

where $\varepsilon > 0$. The corresponding Jacobi field $J_{g,\xi}$ along g is then given by the vector field

$$J_{g,\xi}(t) := \frac{D}{\partial s}\Gamma_{g,\xi}(s, t)\Big|_{s=0}$$

that represents the direction of the displacement of g if x is perturbed in a direction ξ.

We directly obtain $J_{g,\xi}(0) = \xi$, and $J_{g,\xi}(1) = 0$ as well as $J_{g,\xi}(t) \in T_{g(t;x,y)}\mathcal{M}$. Since $\Gamma_{g,\xi}(s, t) = g(t; \gamma_{x,\xi}(s), y)$ we obtain by the chain rule

$$D_x g(t, \cdot, y)[\xi] = \frac{d}{ds} g(t; \gamma_{x,\xi}(s), y)\Big|_{s=0} = \frac{d}{ds}\Gamma_{g,\xi}(s, t)\Big|_{s=0} = J_{g,\xi}(t). \tag{10}$$

Remark. This relation between the derivative of geodesics and Jacobi fields is of high interest on symmetric spaces, where Jacobi fields can be computed in closed form, as summarized in the following Lemma.

Lemma 6. *[22, Prop. 3.5] Let $\mathcal{M}$ be a m-dimensional symmetric Riemannian manifold. Let $g(t; x, y)$, $t \in [0, 1]$, be a geodesic between $x, y \in \mathcal{M}$, $\eta \in T_x\mathcal{M}$ a tangent vector and $\{\xi_1, \dots \xi_m\}$ be an orthonormal basis (ONB) of $T_x\mathcal{M}$ that diagonalizes the curvature operator of $\mathcal{M}$ with eigenvalues κ_ℓ, $\ell = 1, \dots, m$. For details, see Ch. 4.2 and 5 (Ex. 5) of [40]. Let further denote by $\{\Xi_1(t), \dots, \Xi_m(t)\}$ the parallel transported frame of*

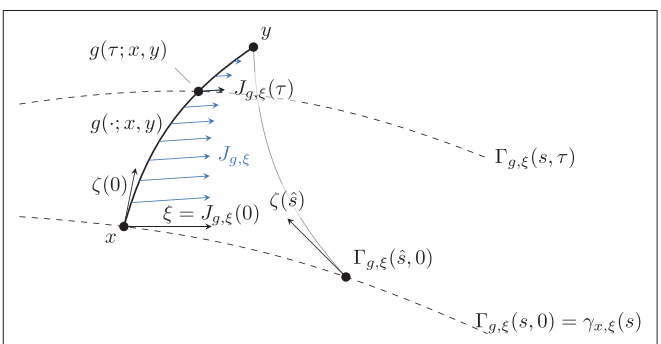

FIGURE 4 | Schematic representation of the variation $\Gamma_{g,\xi}(s, t)$ of a geodesic g w.r.t. the direction $\xi \in T_x\mathcal{M}$. The corresponding Jacobi field along g and in the direction ξ is the vector field $J_{g,\xi}(t) = \frac{\partial}{\partial s}\Gamma_{g,\xi}(s, t)|_{s=0}$.

$\{\xi_1,\dots,\xi_m\}$ along g. Decomposing $\eta = \sum_{\ell=1}^m \eta_\ell \xi_\ell \in T_x\mathcal{M}$, the derivative $D_x g[\eta]$ becomes

$$D_x g(t;x,y)[\eta] = J_{g,\eta}(t) = \sum_{\ell=1}^m \eta_\ell J_{g,\xi_\ell}(t),$$

where the Jacobi field $J_{g,\xi_\ell}\colon \mathbb{R} \to T_{g(t;x,y)}\mathcal{M}$ along g and in the direction ξ_ℓ is given by

$$J_{g,\xi_\ell}(t) = \begin{cases} \frac{\sinh\left(d_g(1-t)\sqrt{-\kappa_\ell}\right)}{\sinh(d_g\sqrt{-\kappa_\ell})}\Xi_\ell(t) & \text{if } \kappa_\ell < 0, \\ \frac{\sin\left(d_g(1-t)\sqrt{\kappa_\ell}\right)}{\sin(\sqrt{\kappa_\ell}d_g)}\Xi_\ell(t) & \text{if } \kappa_\ell > 0, \\ (1-t)\Xi_\ell(t) & \text{if } \kappa_\ell = 0, \end{cases} \tag{11}$$

with $d_g = d(x,y)$ denoting the length of the geodesic $g(t;x,y), t \in [0,1]$.

The Jacobi field of the reversed geodesic $\bar{g}(t) \coloneqq g(t;y,x) = g(1-t;x,y)$ is obtained using the same orthonormal basis and transported frame but evaluated at $s = 1-t$. We thus obtain $D_y g(t;x,y)[\xi_\ell] = D_y g(1-t;y,x)[\xi_\ell] = J_{\bar{g},\xi_\ell}(1-t)$, where

$$J_{\bar{g},\xi_\ell}(1-t) = \begin{cases} \frac{\sinh\left(d_g t\sqrt{-\kappa_\ell}\right)}{\sinh(d_g\sqrt{-\kappa_\ell})}\Xi_\ell(t) & \text{if } \kappa_\ell < 0, \\ \frac{\sin\left(d_g t\sqrt{\kappa_\ell}\right)}{\sin(d_g\sqrt{\kappa_\ell})}\Xi_\ell(t) & \text{if } \kappa_\ell > 0, \\ t\Xi_\ell(t) & \text{if } \kappa_\ell = 0. \end{cases}$$

Note that $T_{g(t)}\mathcal{M} = T_{\bar{g}(1-t)}\mathcal{M}$. Therefore $\Xi_\ell(t) \in T_{g(t)}\mathcal{M}$, $\ell = 1,\dots,m$, is an orthonormal basis for this tangent space.

3.2. Derivative of Coupled Geodesics

Let $\mathcal{M}$ be a symmetric Riemannian manifold. We use the result of Lemma 6 to directly compute the derivative of coupled geodesics, i.e., a function composed of $g_1(t) \coloneqq g(t;x,y)$ and $g_2(t) \coloneqq g(t;g_1(t),z)$. By Definition 5, we have

$$D_x g_2(t)[\eta] = D_{g_1(t)} g(t;\cdot,z)\big[D_x g_1(t)[\eta]\big]$$

and by (10), we obtain

$$D_x g_2(t)[\eta] = J_{g_2, D_x g_1(t)[\eta]}(t),$$

where the variation direction used in the Jacobi field is now the derivative of $g_1(t)$ in direction η. Similarly, we compute the derivative of a reversed coupled geodesic $g_3(t) \coloneqq g(t;z,g_1(t))$ as

$$D_x g_3(t)[\eta] = D_{g_1(t)} g(t;z,\cdot)\big[D_x g_1(t)[\eta]\big] = J_{\bar{g}_3, D_x g_1(t)[\eta]}(1-t).$$

Note that the Jacobi field is reversed here, but that its variation direction is the same as the one of the Jacobi field introduced for $g_2(t)$. In a computational perspective, it means that we can use the same ONB for the derivatives of both g_3 and $\bar{g}_3$ Furthermore, in this case, the variation direction is also computed by a Jacobi field since $D_x g_1(t)[\eta] = J_{g_1,\eta}(t)$.

Finally the derivative of g_2 (resp. g_3) on symmetric spaces is obtained as follows. Let $\{\xi_1^{[1]},\dots,\xi_m^{[1]}\}$ be an ONB of $T_x\mathcal{M}$ for the inner Jacobi field along g_1, and $\{\xi_1^{[2]},\dots,\xi_m^{[2]}\}$ be an ONB of $T_{g_1(t)}\mathcal{M}$ for the outer Jacobi field along g_2 (resp. g_3). As $\eta = \sum_{\ell=1}^m \eta_\ell \xi_\ell^{[1]} \in T_x\mathcal{M}$, and stating $J_{g_1,\xi_\ell^{[1]}}(t) = \sum_{l=1}^m \mu_l \xi_l^{[2]} \in T_{g_1(t)}\mathcal{M}$, the derivative of g_2 (resp. g_3) with respect to x in the direction $\eta \in T_x\mathcal{M}$ reads

$$D_x g_2(t)[\eta] = \sum_{l=1}^m \sum_{\ell=1}^m J_{g_2,\xi_l^{[2]}}(t)\mu_l\eta_\ell, \tag{12}$$

and accordingly for g_3.

3.3. Derivative of a Bézier Curve

Sections 3.1 and 3.2 introduced the necessary concepts to compute the derivative of a general Bézier curve $\beta_K(t;b_0,\dots,b_K)$, as described in Equation (3), with respect to its control points b_j. For readability of the recursive structure investigated in the following, we introduce a slightly simpler notation and the following setting.

Let K be the degree of a Bézier curve $\beta_K(t;b_0,\dots,b_K)$ with the control points $b_0,\dots,b_K \in \mathcal{M}$. We fix $k \in \{1,\dots,K\}$, $i \in \{0,\dots,K-k\}$ and $t \in [0,1]$. We introduce

$$g_i^{[k]}(t) \coloneqq g(t; g_i^{[k-1]}(t), g_{i+1}^{[k-1]}(t)) = \beta_i^{[k]}(t;b_i,\dots,b_{i+k}), \tag{13}$$

for the i^{th} Bézier curve of degree k in the De Casteljau algorithm, and $g_i^{[0]}(t) = b_i$.

Furthermore, given $x \in \{b_i,\dots,b_{i+k}\}$, we denote by

$$\eta_i^{[k]} \coloneqq D_x g_i^{[k]}(t)[\eta], \tag{14}$$

its derivative with respect to one of its control points x in the direction $\eta \in T_x\mathcal{M}$.

Remark. Clearly any other derivative of $g_i^{[k]}$ with respect to $x = b_j$, $j < i$ or $j > i+k$ is zero. In addition we have $\eta_i^{[0]} = D_x g_i^{[0]}[\eta] = \eta$ for $x = b_i$ and zero otherwise.

Theorem 7 (Derivative of a Bézier curve). *Let $k \in \{1,\dots,K\}$ and $i \in \{0,\dots,K-k\}$ be given. The derivative $\eta_i^{[k]} = D_x g_i^{[k]}(t)[\eta]$ of $g_i^{[k]}$ with respect to its control point $x \coloneqq b_j$, $i \le j \le i+k$, and in the direction $\eta \in T_x\mathcal{M}$ is given by*

$$\begin{aligned} \eta_i^{[k]} &\coloneqq D_x g_i^{[k]}(t)[\eta] \\ &= \begin{cases} J_{g_i^{[k]},\eta_i^{[k-1]}}(t) & \text{if } j = i, \\ J_{g_i^{[k]},\eta_i^{[k-1]}}(t) + J_{\bar{g}_i^{[k]},\eta_{i+1}^{[k-1]}}(1-t) & \text{if } i < j < i+k, \\ J_{\bar{g}_i^{[k]},\eta_{i+1}^{[k-1]}}(1-t) & \text{if } j = i+k. \end{cases} \end{aligned}$$

Proof: Let fix $t \in [0,1]$ and $x = b_j$, $i \le j \le i+k$. For readability we set $a \coloneqq g_i^{[k-1]}(t)$, $b \coloneqq g_{i+1}^{[k-1]}(t)$, and $f \coloneqq g_i^{[k]}(t) = g(t;a,b)$. Note that while f depends on the control points $b_i,\dots,b_{i+k}$ and is a Bézier curve of degree k, both a and b are Bézier curves of degree $k-1$. The former does not depend on b_{i+k}, and the latter is independent of b_i.

We prove the claim by induction. For $k = 1$ the function $g_i^{[1]}$ is just a geodesic. The case $i < j < i+1$ does not occur and the

remaining first and third cases follow by the notation introduced for $k = 0$ and Lemma 6.

For $k > 1$ we apply the chain rule (9) to $D_x f[\eta]$ and obtain

$$D_x f[\eta] = D_a f\big[D_x a[\eta]\big] + D_b f\big[D_x b[\eta]\big].$$

Consider the first term $D_a f\big[D_x a[\eta]\big]$ and $j < i + k$. By (10) and the notation from (14), one directly has

$$D_a f\big[\eta_i^{[k-1]}\big] = J_{f,\eta_i^{[k-1]}}(t).$$

For $j = i + k$, clearly $D_x a[\eta] = D_a f[D_x a[\eta]] = 0$, as a does not depend on b_{i+k}.

We proof the second term similarly. For $j > i$, by applying the chain rule and using the reversed Jacobi field formulation of Lemma 6 for the derivative of a geodesic with respect to its end point, we obtain

$$D_b f\big[\eta_{i+1}^{[k-1]}\big] = J_{\bar{f},\eta_{i+1}^{[k-1]}}(1-t).$$

Finally, as $D_x b[\eta] = D_b f[D_x b[\eta]] = 0$ for $x = b_i$, the assumption follows. □

Figure 5 represents one level of the schematic propagation tree to compute the derivative of a Bézier curve.

Example 8 (Quadratic Bézier curve)**.** *Consider the quadratic Bézier curve $\beta_2 \colon [0,1] \to \mathcal{M}$ defined as*

$$\beta_2(t; b_0, b_1, b_2) = g\big(t; g(t; b_0, b_1), g(t; b_1, b_2)\big).$$

Using the notations (13), we have

$$g_0^{[1]}(t) \coloneqq g(t; b_0, b_1), \quad g_1^{[1]}(t) \coloneqq g(t; b_1, b_2),$$
$$g_0^{[2]}(t) \coloneqq g(t; g_0^{[1]}, g_1^{[1]}).$$

The derivative of β_2 at t with respect to b_0 in the direction $\eta \in T_{b_o}\mathcal{M}$, is given by

$$D_{b_0}\beta_2[\eta] = J_{g_0^{[2]},\eta_0^{[1]}}(t), \text{ with } \eta_0^{[1]} \coloneqq J_{g_0^{[1]},\eta}(t).$$

The derivative of β_2 at t with respect to b_2 in the direction $\eta \in T_{b_2}\mathcal{M}$ can be seen as deriving by the first point after inverting the Bezier curve, i.e., looking at $\bar{\beta}_2(t) = \beta_2(1-t)$, hence we have analogously to the first term

$$D_{b_2}\beta_2[\eta] = J_{\bar{g}_0^{[2]},\eta_1^{[1]}}(1-t), \text{ with } \eta_1^{[1]} \coloneqq J_{\bar{g}_1^{[1]},\eta}(1-t).$$

The case $D_{b_1}\beta_2[\eta]$, $\eta \in T_{b_1}\mathcal{M}$, involves a chain rule, where b_1 appears in both $g_0^{[1]}$ (as its end point) and $g_1^{[1]}$ (as its starting point). Using the two intermediate results (or Jacobi fields of geodesics)

$$\eta_0^{[1]} \coloneqq J_{\bar{g}_0^{[1]},\eta}(1-t) \quad \text{and} \quad \eta_1^{[1]} \coloneqq J_{g_1^{[1]},\eta}(t),$$

we obtain

$$D_{b_1}\beta_2[\eta] = J_{g_0^{[2]},\eta_0^{[1]}}(t) + J_{\bar{g}_0^{[2]},\eta_1^{[1]}}(1-t).$$

Example 9 (Cubic Bézier curve)**.** *Consider the cubic Bézier curve $\beta_3 \colon [0,1] \to \mathcal{M}$ defined as*

$$\beta_3(t; b_0, b_1, b_2, b_3) = g\big(t; \beta_2(t; b_0, b_1, b_2), \beta_2(t; b_1, b_2, b_3)\big).$$

As in Example 8, we use the notations (13) and define

$$g_j^{[1]}(t) \coloneqq g(t; b_j, b_{j+1}), \qquad j = 0, 1, 2,$$
$$g_j^{[2]}(t) \coloneqq g(t; g_j^{[1]}, g_{j+1}^{[1]}), \qquad j = 0, 1, \text{ and}$$
$$g_0^{[3]}(t) \coloneqq g(t; g_0^{[2]}, g_1^{[2]}).$$

*The derivation of β_3 with respect to b_0 or b_3 follows the same structure as in Example 8. The case of $D_{b_1}\beta_3[\eta]$, however, requires two chain rules. The needed Jacobi fields follow the tree structure shown in **Figure 6B**: given $\eta \in T_{b_1}\mathcal{M}$, we define at the first recursion step*

$$\eta_0^{[1]} \coloneqq J_{\bar{g}_0^{[1]},\eta}(1-t), \qquad \eta_1^{[1]} \coloneqq J_{g_1^{[1]},\eta}(t),$$

and at the second recursion step

$$\eta_0^{[2]} \coloneqq J_{g_0^{[2]},\eta_0^{[1]}}(t) + J_{\bar{g}_0^{[2]},\eta_1^{[1]}}(1-t), \qquad \eta_1^{[2]} \coloneqq J_{g_1^{[2]},\eta_1^{[1]}}(t).$$

Note that both $\eta_0^{[2]}$ and $\eta_1^{[2]}$ are actually the derivative of $\beta_2(t; b_0, b_1, b_2)$ and $\beta_2(t; b_1, b_2, b_3)$, respectively, with respect to b_1 and direction $\eta \in T_{b_1}\mathcal{M}$. Finally we have

$$D_{b_1}\beta_3[\eta] = J_{g_0^{[3]},\eta_0^{[2]}}(t) + J_{\bar{g}_0^{[3]},\eta_1^{[2]}}(1-t).$$

The case of $D_{b_2}\beta_3[\eta]$ is obtained symmetrically, with arguments similar to b_1.

Note that computing the Jacobi fields involved just follows the same decomposition tree as the De Casteljau algorithm (**Figure 6A**).

3.4. Joining Segments and Deriving the Gradient

In this subsection we derive the differential of a composite Bézier curve $\mathbf{B}(t)$ consisting of n segments and take the C^1 conditions into account. We simplify the notations from section 2.1 and set the degree fixed to $K_i = K$ for all segments, e.g., $K = 3$ for a cubic composite Bézier curve. Then the control points are b_j^i, $j = 0, \ldots, K$, $i = 0, \ldots, n-1$. We further denote by $p_i = b_K^{i-1} = b_0^i$, $i = 1, \ldots, n-1$ the common junction point of the segments and p_0 and p_n the start and end points, respectively. For ease of notation we denote by $b_i^- = b_{K-1}^i$ and $b_i^+ = b_1^i$, $i = 1, \ldots, n-1$, the two points needed for differentiability (C^1) condition investigation, cf. **Figure 1** for an illustration of the framework on $\mathcal{M} = \mathbb{R}$, with $K = 3$.

One possibility to enforce the C^1 condition (4) is to include it into the composite Bézier curve by replacing b_i^+ with

$$b_i^+ = g(2; b_i^-, p_i), \quad i = 1, \ldots, n-1. \tag{15}$$

This way both the directional derivatives of $\mathbf{B}(t)$ with respect to b_i^- and p_i change due to a further (most inner) chain rule.

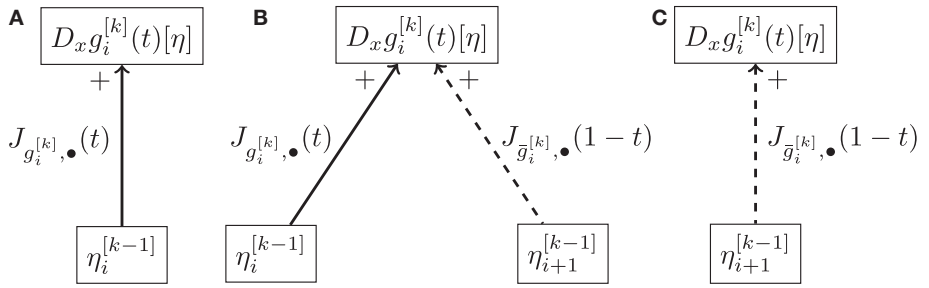

FIGURE 5 | Schematic representation of the cases where elements compose the chained derivative of the *i*th Bézier curve of order *k* in the De Casteljau algorithm. The solid line represents a Jacobi field along $g_i^{[k]}$, while the dashed one represents a *reversed* Jacobi field. **(A)** Represents the case $x = b_i$, **(B)** represents the cases where $x \in \{b_{i+1}, \dots, b_{i+k+1}\}$, and **(C)** represents the case $x = b_{i+k}$.

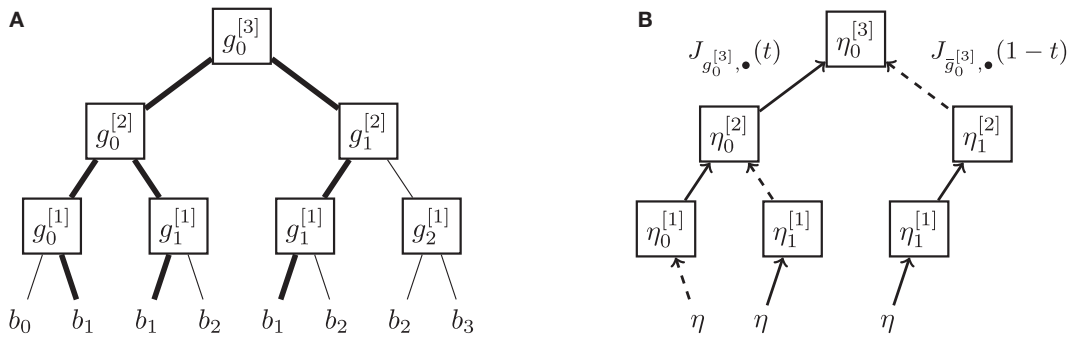

FIGURE 6 | Construction and derivation tree of a Bézier curve $\beta_3(t; b_0, b_1, b_2, b_3)$. The derivative with respect to a variable b_i is obtained by a recursion of Jacobi fields added at each leaf of the tree. **(A)** Tree-representation of the construction of a cubic Bezier curve. The thick line tracks the propagation of b_1 within the tree. **(B)** Tree-representation of the construction of $\eta_0^{[3]} := D_{b_1}\beta_3[\eta]$. The solid lines are Jacobi fields while dashed lines are *reversed* Jacobi fields.

Lemma 10 (Derivative of a composite Bézier curve with C^1 condition)**.** *Let* $\mathbf{B}$ *be a composite Bézier curve and* p_i, b_i^+, b_i^- *introduced as above. Replacing* $b_i^+ = g(2; b_i^-, p_i)$ *eliminates that variable from the composite Bezier curve and keeps the remaining differentials unchanged, except the following which now read*

$$D_{b_i^-}\mathbf{B}(t)[\eta] = \begin{cases} D_{b_i^-}\beta_K(t-i+1; \\ \quad p_{i-1}, b_{i-1}^+, \dots, \cdot, p_i)[\eta] & t \in (i-1, i], \\ D_{b_i^+}\beta_K(t-i; \quad p_i, b_i^+, \dots, \cdot, p_{i+1}) \\ \quad \left[D_{b_i^-} g(2; \cdot, p_i)[\eta]\right] & t \in (i, i+1], \end{cases}$$

and

$$D_{p_i}\mathbf{B}(t)[\eta] = \begin{cases} D_{p_i}\beta_K\,(\,t-i+1; p_{i-1}, \\ \quad b_{i-1}^+, \dots, b_i^-, \cdot\,)\,[\eta] & t \in (i-1, i], \\ D_{p_i}\beta_K(t-i; \cdot, \\ \quad b_i^+, \dots, b_{i+1}^-, p_{i+1})[\eta] \\ +D_{b_i^+}\beta_K(t-i; p_i, \cdot, \dots, \\ \quad b_{i+1}^-, p_{i+1})\left[D_{p_i} g(2; b_i^-, \cdot)[\eta]\right] & t \in (i, i+1]. \end{cases}$$

In both cases, the first interval includes $i - 1 = 0$*, when* $i = 1$.

Proof: Both first cases are from the derivation of a Bézier curve as before, for both second cases replacing $b_i^+ = g(2; b_i^-, p_i)$ yields one (for p_i additional) term.

We now derive the gradient of the objective function (7). We introduce the abbreviation $\mathbf{B}_i = \mathbf{B}(t_i) \in \mathcal{M}$, and $\mathrm{d}_{2,i}^2 = \mathrm{d}_2^2(\mathbf{B}_{i-1}, \mathbf{B}_i, \mathbf{B}_{i+1})$.

Theorem 11. *Let* $\mathcal{M}$ *be a m-dimensional manifold, x be one of the control points of a composite Bézier curve* $\mathbf{B}$*, and* $\{\xi_1, \dots, \xi_m\}$ *be a corresponding orthonormal basis (ONB) of* $T_x\mathcal{M}$*. The gradient* $\nabla_{\mathcal{M},x}A(\mathrm{b})$ *of the discretized mean squared acceleration* $A(\mathbf{b})$ *of* $\mathbf{B}$*, w.r.t. x, discretized at* $N+1$ *equispaced times* $t_0, \dots, t_N$ *is given by*

$$\nabla_{\mathcal{M},x}A(\mathbf{b}) = \frac{1}{\Delta t^3}\sum_{\ell=1}^{m}\sum_{i=1}^{N-1}\sum_{j=i-1}^{i+1}\langle \nabla_{\mathcal{M}}\mathrm{d}_{2,i}^2, D_x\mathbf{B}_j[\xi_\ell]\rangle_{\mathbf{B}_j}\xi_\ell.$$

Proof: As $\nabla_{\mathcal{M},x}A(\mathbf{b}) \in T_x\mathcal{M}$, we seek for the coefficients $a_\ell := a_\ell(x)$ such that

$$\nabla_{\mathcal{M}} f(x) = \sum_{\ell=1}^{m} a_\ell \xi_\ell. \tag{16}$$

Therefore, for any tangential vector $\eta := \sum_{\ell=1}^{m} \eta_\ell \xi_\ell \in T_x\mathcal{M}$, we have,

$$\langle \nabla_{\mathcal{M},x} A(\mathbf{b}), \eta\rangle_x = \sum_{\ell=1}^{m} a_\ell \eta_\ell. \tag{17}$$

By definition of A and Definition 4 this yields

$$\langle \nabla_{\mathcal{M},x} A(\mathbf{b}), \eta\rangle_x = \frac{1}{\Delta t^3} \sum_{i=1}^{N-1} \langle \nabla_{\mathcal{M}} \mathrm{d}^2_{2,i}(x), \eta\rangle_x = \frac{1}{\Delta t^3} \sum_{i=1}^{N-1} D_x \mathrm{d}^2_{2,i}[\eta]. \tag{18}$$

We compute $D_x \mathrm{d}^2_{2,i}[\eta]$ using the chain rule (Definition 5) as

$$D_x \mathrm{d}^2_{2,i}[\eta] = \sum_{j=i-1}^{i+1} D_{\mathbf{B}_j} \mathrm{d}^2_{2,i}\big[D_x \mathbf{B}_j[\eta]\big], \tag{19}$$

which, by Definition 4, again becomes

$$D_{\mathbf{B}_j} \mathrm{d}^2_{2,i}\big[D_x \mathbf{B}_j[\eta]\big] = \langle \nabla_{\mathcal{M}} \mathrm{d}^2_{2,i}, D_x \mathbf{B}_j[\eta]\rangle_{\mathbf{B}_j}.$$

The term on the left of the inner product is given in Bačák et al. [22, section 3] and the right term is given in section 3.3. While the former can be computed using Jacobi fields and a logarithmic map, the latter is the iteratively coupling of Jacobi fields. Furthermore, the differential $D_x \mathbf{B}_j[\eta]$ can be written as

$$D_x \mathbf{B}_j[\eta] = \sum_{\ell=1}^{m} \eta_\ell D_x \mathbf{B}_j[\xi_\ell] \in T_{\mathbf{B}_j}\mathcal{M}.$$

Hence, we obtain

$$D_{\mathbf{B}_j} \mathrm{d}^2_{2,i}\big[D_x \mathbf{B}_j[\eta]\big] = \sum_{\ell=1}^{m} \eta_\ell \langle \nabla_{\mathcal{M}} \mathrm{d}^2_{2,i}, D_x \mathbf{B}_j[\xi_\ell]\rangle_{\mathbf{B}_j},$$

and by (17), (18), and (19), it follows

$$\langle \nabla_{\mathcal{M},x} A(\mathbf{b}), \eta\rangle_x = \frac{1}{\Delta t^3} \sum_{\ell=1}^{m} \eta_\ell \sum_{i=1}^{N-1} \sum_{j=i-1}^{i+1} \langle \nabla_{\mathcal{M}} \mathrm{d}^2_{2,i}, D_x \mathbf{B}_j[\xi_\ell]\rangle_{\mathbf{B}_j},$$

which yields the assertion (16).

4. APPLICATION TO THE FITTING PROBLEM

The fitting problem has been tackled different ways this last decades. The approach with Bézier curves is more recent, and we refer to Absil et al. [35], Arnould et al. [37], and Gousenbourger et al. [38] for a detailed overview of these methods.

In this section, we present the numerical framework we use in order to fit a composite Bézier curve $\mathbf{B}$ to a set of data points $d_0, \ldots, d_n \in \mathcal{M}$ associated with time-parameters $t_0, \ldots, t_n$, such that we meet (1). For the sake of simplicity, we limit the study to the case where $t_i = i$, $i = 0, \ldots, n$. Therefore, the fitting problem (1) becomes

$$\min_{\mathbf{b}\in\Gamma_{\mathbf{B}}} E_\lambda(\mathrm{b}) := \min_{\mathrm{b}\in\Gamma_{\mathbf{B}}} \int_{t_0}^{t_n} \left\| \frac{\mathrm{D}^2 \mathbf{B}(t)}{\mathrm{d}t^2} \right\|^2_{\mathbf{B}(t)} \mathrm{d}t + \frac{\lambda}{2} \sum_{i=0}^{n} \mathrm{d}^2(p_i, d_i), \tag{20}$$

where $\Gamma_{\mathbf{B}} \in \mathcal{M}^M$ is the set of the M control points of $\mathbf{B}$. Remark that, compared to (1), the optimization is now done on the product manifold $\mathcal{M}^M$. Furthermore, the fitting term now implies a distance between d_i and p_i, as $\mathbf{B}(t_i) = p_i$.

The section is divided in three parts: the product manifold $\mathcal{M}^M$ is defined in section 4.1, where the contribution of the fitting term in the gradient of E is also presented. Then, we propose an efficient algorithm to compute the gradient of the discretized MSA, based on so-called *adjoint Jacobi fields*. We finally shortly mention the gradient descent algorithm we use as well as the involved Armijo rule.

4.1. Fitting and Interpolation

Let us clarify the set $\Gamma_{\mathbf{B}} \in \mathcal{M}^M$ from (20). We will explicitly present the vector b and state its size M. The set $\Gamma_{\mathbf{B}}$ is the set of the M remaining free control points to optimize, when the $\mathcal{C}^1$ continuity constraints are imposed. We distinguish two cases: (i) the fitting case, that corresponds to formulae presented in section 3, and (ii) the interpolation case ($\lambda \to \infty$) where the constraint $d_i = p_i$ is imposed as well.

For a given composite Bézier curve $\mathbf{B}\colon [0, n] \to \mathcal{M}$ consisting of a Bézier curve of degree K on each segment, and given the C^1 conditions (4), the segments are determined by the points

$$\mathbf{b} = (p_0, b_0^+, b_2^0, \ldots, b_{K-2}^0, b_1^-, p_1, \ldots, b_n^-, p_n) \in \mathcal{M}^M. \tag{21}$$

We investigate the length of M. First, b_i^+ is given by p_i and b_i^- via (4). Second, for the segments $i = 2, \ldots, n$ we can further omit p_{i-1} since the value also occurs in the segment $i-1$ as last entry. Finally, the first segment contains the additional value of b_0^+ that is not fixed by C^1 constraints. The first segment is thus composed of $K+1$ control points, while the $n-1$ remaining segments are determined by $K-1$ points. In total we obtain $M = n(K-1)+2$ control points to optimize.

Minimizing $A(\mathbf{b})$ alone leads to the trivial solution, for any set of control points $\mathbf{b} = (x, \ldots, x)$, $x \in \mathcal{M}$, and this is why the fitting term from (20) is important.

Fitting ($0 < \lambda < \infty$). If the segment start and end points are obstructed by noise or allowed to move, we employ a fitting scheme to balance the importance given to the data points $d_0, \ldots, d_n$. Equation (20) reads

$$\operatorname*{argmin}_{\mathbf{b}\in\Gamma_{\mathbf{B}}} \tilde{A}(\mathbf{b}), \quad \tilde{A}(\mathbf{b}) := A(\mathbf{b}) + \frac{\lambda}{2} \sum_{i=0}^{n} d^2_{\mathcal{M}}(d_i, p_i), \tag{22}$$

where $\lambda \in \mathbb{R}^+$ sets the priority to either the data term (large λ) or the mean squared acceleration (small λ) within the minimization. The gradient of the data term is given in Karcher [41], and the gradient of $\tilde{A}$ is given by

$$\nabla_{\mathcal{M}^M,x} \tilde{A}(\mathbf{b}) = \begin{cases} \nabla_{\mathcal{M}^M,x} A(\mathbf{b}) - \lambda \log_{p_i} d_i & \text{if } x = p_i,\ i = 0, \ldots, n, \\ \nabla_{\mathcal{M}^M,x} A(\mathbf{b}) & \text{otherwise.} \end{cases}$$

Interpolation ($\lambda \to \infty$). For interpolation we assume that the start point p_{i-1} and end point p_i of the segments $i = 1, \dots, n$ are fixed to given data d_{i-1} and $d_i \in \mathcal{M}$, respectively. The optimization of the discrete mean squared acceleration $A(\mathbf{b})$ reads

$$\operatorname*{argmin}_{\mathbf{b}\in\Gamma_\mathbf{B}} A(\mathbf{b}) \quad \text{s. t. } p_i = d_i,\ i = 0, \dots, n. \tag{23}$$

Since the p_i are fixed by constraint, they can be omitted from the vector $\mathbf{b}$. We obtain

$$\mathbf{b} = (b_0^+, b_2^0, \dots, b_1^-, b_2^1, \dots, b_n^-) \in \mathcal{M}^{M'}$$

Since there are $n+1$ additional constraints, the minimization is hence performed on the product manifold $\mathcal{M}^{M'}$, $M' = M - (n+1) = n(K-2)+1$.

4.2. Adjoint Jacobi Fields

In the Euclidean space $\mathbb{R}^m$, the adjoint operator T^* of a linear bounded operator $T\colon \mathbb{R}^m \to \mathbb{R}^q$ is the operator fulfilling

$$\langle T(x), y\rangle_{\mathbb{R}^q} = \langle x, T^*(y)\rangle_{\mathbb{R}^m}, \quad \text{for all } x \in \mathbb{R}^m,\ y \in \mathbb{R}^q.$$

The same can be defined for a linear operator $S\colon T_x\mathcal{M} \to T_y\mathcal{M}$, $x, y \in \mathcal{M}$, on a m-dimensional Riemannian manifold $\mathcal{M}$. The adjoint operator $S^*\colon T_y\mathcal{M} \to T_x\mathcal{M}$ satisfies

$$\langle S(\eta), \nu\rangle_y = \langle \eta, S^*(\nu)\rangle_x, \quad \text{for all } \eta \in T_x\mathcal{M},\ \nu \in T_y\mathcal{M}.$$

We are interested in the case where S is the differential operator D_x of a geodesic $F(x) = g(t; x, y)$ for some fixed $t \in \mathbb{R}$ and $y \in \mathcal{M}$. The differential $D_xF\colon T_x\mathcal{M} \to T_{F(x)}\mathcal{M}$ can be written as

$$D_xF[\eta] = J_{g,\eta}(t) = \sum_{\ell=1}^{m} \langle \eta, \xi_\ell\rangle_x \alpha_\ell\, \Xi_\ell(t),$$

where α_ℓ are the coefficients of the Jacobi field (11), and ξ_ℓ, $\Xi_\ell(t)$ are given as in Lemma 6. To derive the *adjoint differential* $(D_xF)^*\colon T_{F(x)}\mathcal{M} \to T_x\mathcal{M}$ we observe that for any $\nu \in T_{F(x)}\mathcal{M}$ we have

$$\begin{aligned}\langle D_xF[\eta], \nu\rangle_{F(x)} &= \sum_{\ell=1}^{m} \langle \eta, \xi_\ell\rangle_x \alpha_\ell \langle \Xi_\ell(t), \nu\rangle_{F(x)} \\ &= \Big\langle \eta, \sum_{\ell=1}^{m} \langle \Xi_\ell(t), \nu\rangle_{F(x)} \alpha_\ell \xi_\ell \Big\rangle_x.\end{aligned}$$

Hence the adjoint differential is given by

$$(D_xF)^*[\nu] = \sum_{\ell=1}^{m} \langle \nu, \Xi_\ell(t)\rangle_{F(x)} \alpha_\ell \xi_\ell, \qquad \nu \in T_{F(x)}\mathcal{M}.$$

We introduce the *adjoint Jacobi field* $J^*_{F,\nu}\colon \mathbb{R} \to T_x\mathcal{M}$, $\nu \in T_{F(x)}\mathcal{M}$ as

$$J^*_{F,\nu}(t) = \sum_{\ell=1}^{m} \langle \nu, \Xi_\ell(t)\rangle_{F(x)} J^*_{F,\Xi_\ell(t)}(t) = \sum_{\ell=1}^{m} \langle \nu, \Xi_\ell(t)\rangle_{F(x)} \alpha_\ell \xi_\ell.$$

Note that evaluating the adjoint Jacobi field J^* involves the same transported frame $\{\Xi_1(t), \dots \Xi_m(t)\}$ and the same coefficients α_ℓ as the Jacobi field J, which means that the evaluation of the adjoint is in no way inferior to the Jacobi field itself.

The adjoint D^* of the differential is useful in particular, when computing the gradient $\nabla_\mathcal{M}(h \circ F)$ of the composition of $F\colon \mathcal{M} \to \mathcal{M}$ with $h\colon \mathcal{M} \to \mathbb{R}$. Setting $y := F(x) \in \mathcal{M}$, we obtain for any $\eta \in T_x\mathcal{M}$ that

$$\begin{aligned}\langle \nabla_{\mathcal{M},x}(h\circ F)(x), \eta\rangle_x &= D_x(h\circ F)[\eta] \\ &= D_{F(x)}h\big[D_xF[\eta]\big] \\ &= \langle \nabla_{\mathcal{M},y}h(y), D_xF[\eta]\rangle_y \\ &= \Big\langle (D_xF)^*\big[\nabla_{\mathcal{M},y}h(y)\big], \eta\Big\rangle_x.\end{aligned}$$

Especially for the evaluation of the gradient of the composite function $h \circ F$ we obtain

$$\nabla_{\mathcal{M},x}(h\circ F)(x) = (D_xF)^*\big[\nabla_{\mathcal{M},y}h(y)\big] = J^*_{F,\nabla_{\mathcal{M},y}h(y)}(t).$$

The main advantage of this technique appears in the case of composite functions, i.e., of the form $h\circ F_1\circ F_2$ (the generalization to composition with K functions is straightforward). The gradient $\nabla_{\mathcal{M},x}(h \circ F_1 \circ F_2)(x)$ now reads,

$$\begin{aligned}\nabla_{\mathcal{M},x}(h\circ F_1\circ F_2)(x) &= (D_xF_2)^*\big[\nabla_{\mathcal{M},y_2}h\circ F_1(y_2)\big] \\ &= J^*_{F_2,\nabla_{\mathcal{M},y_2}h\circ F_1(y_2))}(t).\end{aligned}$$

The recursive computation of $\eta^{[3]} = \nabla_{\mathcal{M},x}h(x)$ is then given by the following algorithm

$$\begin{aligned}\eta^{[1]} &= \nabla_{\mathcal{M},y_1}h(y_1), \\ \eta^{[2]} &= J^*_{F_1,\eta^{[1]}}(t), \\ \eta^{[3]} &= J^*_{F_2,\eta^{[2]}}(t).\end{aligned}$$

Example 12. *For $h = d_2^2\colon \mathcal{M}^3 \to \mathbb{R}$ we know $\nabla_{\mathcal{M}^3}h$ by Lemma 3.1 and Lemma 3.2 from Bačák et al. [22]. Let $t_1, t_2, t_3 \in [0,1]$ be time points, and $\mathbf{b} \in \mathcal{M}^M$ be a given (sub)set of the control points of a (composite) Bézier curve $\mathbf{B}$. We define $F\colon \mathcal{M}^M \to \mathcal{M}^3$, $\mathbf{b} \mapsto F(\mathrm{b}) = (\mathbf{B}(t_1), \mathbf{B}(t_2), \mathbf{B}(t_3))$ as the evaluations of $\mathbf{B}$ at the three given time points. The composition $h \circ F$ hence consists of (in order of evaluation) the geodesic evaluations of the De Casteljau algorithm, the mid point function for the first and third time point and a distance function.*

The recursive evaluation of the gradient starts with the gradient of the distance function. Then, for the first and third arguments, a mid point Jacobi field is applied. The result is plugged into the last geodesic evaluated within the De Casteljau "tree" of geodesics. At each geodesic, a tangent vector at a point $g(t_j; a, b)$ is the input for two adjoint Jacobi fields, one mapping to $T_a\mathcal{M}$, the other to $T_b\mathcal{M}$. This information is available throughout the recursion steps anyways. After traversing this tree backwards, one obtains the required gradient of $h \circ F$.

Note also that even the differentiability constraint (4) yields only two further (most outer) adjoint Jacobi fields,

Algorithm 1 Gradient descent algorithm on a manifold $N = \mathcal{M}^M$

Input. $F\colon \mathcal{N} \to \mathbb{R}$, its gradient $\nabla_{\mathcal{N}} F$, $x^{(0)} \in \mathcal{N}$, step sizes $s_k > 0, k \in \mathbb{N}$.
Output: $\hat{x} \in \mathcal{N}$
$k \leftarrow 0$
repeat
  Perform a gradient descent step $x^{(k+1)} := \exp_{x^{(k)}}\left(-s_k \nabla_{\mathcal{N}} F(x^{(k)})\right)$
  $k \leftarrow k+1$
until a stopping criterion is reached
return $\hat{x} := x^{(k)}$

namely $J^*_{g(2,b_i^-,p_i),\nabla_{\mathcal{M},b_i^+}\mathbf{B}(t_j)}(2)$ and $J^*_{\tilde{g}(2,b_i^-,p_i),\nabla_{\mathcal{M},b_i^+}\mathbf{B}(t_j)}(2)$. They correspond to variation of the start point b_i^- and the end point p_i, respectively as stated in (10).

4.3. A Gradient Descent Algorithm

To address (22) or (23), we use a gradient descent algorithm, as described in Absil et al. [10, Ch. 4]. For completeness the algorithm is given in **Algorithm 1**.

The step sizes are given by the Armijo line search condition presented in Absil et al. [10, Def. 4.2.2]. Let $\mathcal{N}$ be a Riemannian manifold, $x = x^{(k)} \in \mathcal{N}$ be an iterate of the gradient descent, and $\beta, \sigma \in (0,1), \alpha > 0$. Let m be the smallest positive integer such that

$$F(x) - F\big(\exp_x(-\beta^m \alpha \nabla_{\mathcal{N}} F(x))\big) \geq \sigma \beta^m \alpha \|\nabla_{\mathcal{N}} F(x)\|_x. \quad (24)$$

We set the step size to $s_k := \beta^m \alpha$ in **Algorithm 1**.

As a stopping criterion we use a maximal number $k_{\max}$ of iterations or a minimal change per iteration $d_{\mathcal{N}}(x^k, x^{k+1}) < \varepsilon$. In practice, this last criterion is matched first.

The gradient descent algorithm converges to a critical point if the function F is convex [10, sections 4.3 and 4.4]. The mid-points model (6) posesses two advantages: (i) the complete discretized MSA (7) consists of (chained) evaluations of geodesics and a distance function, and (ii) it reduces to the classical second order differences on the Euclidean space. However, this model is not convex on general manifolds. An example is given in the arXiv preprint (version 3) of Bačák et al. [22][1], Remark 4.6. Another possibility (also reducing to classical second order differences in Euclidean space) is the so-called Log model (see e.g., [42])

$$d_{2,\mathrm{Log}}[x,y,z] = \|\log_y x + \log_y z\|_y.$$

The (merely technical) disadvantage of the Log model is that the computation of the gradient involves further Jacobi fields than the one presented above, namely to compute the differentials of the logarithmic map both with respect to its argument $D_x \log_y x$ as well as its base point $D_y \log_y x$. Still, these can be given in closed form for symmetric Riemannian manifolds [23, Th. 7.2][43, Lem. 2.3]. To the best of our knowledge, the joint convexity of the Log model in x, y and z is still an open question.

[1] See arxiv.org/abs/1506.02409v3

5. EXAMPLES

In this section, we provide several examples of our algorithm applied to the fitting problem (20).

We validate it first on the Euclidean space and verify that it retrieves the natural cubic smoothing spline. We then present examples on the sphere $\mathbb{S}^2$ and the special orthogonal group SO(3). We compare our results with the fast algorithm of Arnould et al. [37], generalized to fitting, and the so-called blended cubic splines from Gousenbourger et al. [38]. The control points of the former are obtained by generalizing the optimal Euclidean conditions of (20) (in $\mathbb{R}^m$, this is a linear system of equations) to the manifold setting; the curve is afterwards reconstructed by a classical De Casteljau algorithm. In the latter, the curve is obtained as a blending of solutions computed on carefully chosen tangent spaces, i.e., Euclidean spaces.

We will show in this section that the proposed method performs as good as the existing methods when the data is from the Eulidean space (section 5.1). This also means that all methods work equally well whenever the data points on the manifold are local enough. However, we will show by the other examples (sections 5.2 and 5.3) that our proposed method outperforms the others whenever the data points are spread out on the manifold.

The following examples were implemented in MVIRT [12][2], and the comparison implementations from Arnould et al. [37] and Gousenbourger et al. [38] use Manopt [11][3]. Note that both toolboxes use a very similar matrix structure and are implemented in Matlab, such that the results can directly be compared.

5.1. Validation on the Euclidean Space

As a first example, we perform a minimization on the Euclidean space $\mathcal{M} = \mathbb{R}^3$. A classical result on $\mathbb{R}^m$ is that the curve γ minimizing (1) is the natural ($\mathcal{C}^2$) cubic spline when Γ is a Sobolev space $H^2(t_0, t_n)$. We compare our approach to the tangential linear system approach derived in Arnould et al. [37] in order to validate our model. We use the following data points

$$d_0 = \begin{bmatrix}0\\0\\1\end{bmatrix},\ d_1 = \begin{bmatrix}0\\-1\\0\end{bmatrix},\ d_2 = \begin{bmatrix}-1\\0\\0\end{bmatrix},\ d_3 = \frac{1}{\sqrt{82}}\begin{bmatrix}0\\-1\\-9\end{bmatrix}, \quad (25)$$

as well as the control points $p_i = d_i$, and

$$\begin{aligned} b_0^+ &= \exp_{p_0} \frac{\pi}{8\sqrt{2}}\begin{bmatrix}1\\-1\\0\end{bmatrix}, & b_1^+ &= \exp_{p_1} -\frac{\pi}{2\sqrt{2}}\begin{bmatrix}1\\0\\1\end{bmatrix}, \\ & & b_1^- &= g(2; b_1^+, p_1), \\ b_2^+ &= \exp_{p_2} \frac{\pi}{2\sqrt{2}}\begin{bmatrix}0\\1\\-1\end{bmatrix}, & b_3^- &= \exp_{p_3} \frac{\pi}{8}\begin{bmatrix}-1\\0\\0\end{bmatrix}, \\ & & b_2^- &= g(2; b_2^+, p_2), \end{aligned} \quad (26)$$

[2] Open source, available at http://ronnybergmann.net/mvirt/
[3] Open source, available at http://www.manopt.org

where the exponential map and geodesic on $\mathbb{R}^3$ are actually the addition and line segments, respectively. Note that, by construction, the initial curve is continuously differentiable but (obviously) does not minimize (1). The parameter λ is set to 50. The MSA of this initial curve $\tilde{A}(\mathbf{b})$ is approximately 18.8828.

The data points are given to the algorithm of Arnould et al. [37] to reconstruct the optimal Bézier curve $\mathbf{B}(t)$, which is the natural $\mathcal{C}^2$ cubic spline. The result is shown in **Figure 7A** together with the first order differences along the curve in **Figure 7B** as a dotted curve.

The set of control points $(p_0, b_0^+, \dots, b_3^-, p_3)$ is optimized with our proposed method. We discretize the second order difference using $N = 1600$ points. The resulting curve and the first order difference plots are also obtained using these sampling values and a first order forward difference. The gradient descent algorithm from **Algorithm 1** employs the Armijo rule (24) setting $\beta = \frac{1}{2}$, $\sigma = 10^{-4}$, and $\alpha = 1$. The stopping criteria are $\sum_{i=1}^{1600} d_{\mathcal{M}}(x_i^{(k)}, x_i^{(k+1)}) < \epsilon = 10^{-15}$ or $\|\nabla_{\mathcal{M}^n}\tilde{A}\|_2 < 10^{-9}$, and the algorithm stops when one of the two is met. For this example, the first criterion stopped the algorithm, while the norm of the gradient was of magnitude 10^{-6}.

Both methods improve the initial functional value of $\tilde{A}(\mathbf{b}) \approx 18.8828$ to a value of $\tilde{A}(\mathbf{b}_{\min}) \approx 4.981218$. Both the linear system approach and the gradient descent perform equally. The difference of objective value is 2.4524×10^{-11} smaller for the gradient descent, and the maximal distance of any sampling point of the resulting curves is of size 4.3×10^{-7}. Hence, in the Euclidean space, the proposed gradient descent yields the natural cubic spline, as one would expect.

5.2. Examples on the Sphere $\mathbb{S}^2$

Validation on a geodesic. As a second example with a known minimizer, we consider the manifold $\mathcal{M} = \mathbb{S}^2$, i.e., the two-dimensional unit sphere embedded in $\mathbb{R}^3$, where geodesics are great arcs. We use the data points

$$d_0 = \begin{bmatrix}0\\0\\1\end{bmatrix}, \quad d_1 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \quad d_2 = \begin{bmatrix}0\\0\\-1\end{bmatrix}$$

aligned on the geodesic connecting the north pole p_0 and the south pole p_2, and running through a point p_1 on the equator. We define the control points of the cubic Bézier curve as follows:

$$x_0 = \frac{1}{\sqrt{6}}\begin{bmatrix}1\\1\\2\end{bmatrix}, \quad b_0^+ = \exp_{p_0}(3\log_{p_0}(x_0)),$$
$$b_1^- = \frac{1}{\sqrt{6}}\begin{bmatrix}1\\2\\1\end{bmatrix},$$
$$x_2 = \frac{1}{\sqrt{6}}\begin{bmatrix}-1\\1\\-2\end{bmatrix}, \quad b_1^+ = \frac{1}{\sqrt{6}}\begin{bmatrix}-1\\2\\-1\end{bmatrix},$$
$$b_2^- = \exp_{p_2}(\tfrac{1}{3}\log_{p_0}(x_2)),$$

where x_0 and x_2 are temporary points and $p_i = d_i$. We obtain two segments smoothly connected since $\log_{p_1} b_1^- = -\log_{p_1} b_1^+$. The original curve is shown in **Figure 8A**, where especially the tangent vectors illustrate the different speed at p_i.

The control points are optimized with our interpolating model, i.e., we fix the start and end points p_0, p_1, p_2 and minimize $A(\mathrm{b})$.

The curve, as well as the second and first order differences, is sampled with $N = 201$ equispaced points. The parameters of the Armijo rule are again set to $\beta = \frac{1}{2}$, $\sigma = 10^{-4}$, and $\alpha = 1$. The stopping criteria are slightly relaxed to 10^{-7} for the distance and 10^{-5} for the gradient, because of the sines and cosines involved in the exponential map.

The result is shown in **Figure 8B**. Since the minimizer is the geodesic running from p_0 to p_2 through p_1, we measure the perfomance first by looking at the resulting first order difference, which is constant, as can be seen in **Figure 8C**. As a second validation, we observe that the maximal distance of the resulting curve to the geodesic is of 2.2357×10^{-6}. These evaluations again validate the quality of the gradient descent.

Effect of the data term. As a third example we investigate the effect of λ in the fitting model. We consider the data points

$$d_0 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^{\mathrm{T}}, \quad d_1 = \begin{bmatrix}0 & -1 & 0\end{bmatrix}^{\mathrm{T}}, \quad d_2 = \begin{bmatrix}-1 & 0 & 0\end{bmatrix}^{\mathrm{T}},$$
$$d_3 = \begin{bmatrix}0 & 0 & -1\end{bmatrix}^{\mathrm{T}},$$

as well as the control points $p_i = d_i$, and

$$b_0^+ = \exp_{p_0}(\frac{\pi}{8\sqrt{2}}\begin{bmatrix}1 & -1 & 0\end{bmatrix}^{\mathrm{T}}), \quad b_1^+ = \exp_{p_1}(-\frac{\pi}{4\sqrt{2}}\begin{bmatrix}-1 & 0 & 1\end{bmatrix}^{\mathrm{T}}),$$
$$b_2^+ = \exp_{p_2}(\frac{\pi}{4\sqrt{2}}\begin{bmatrix}0 & 1 & -1\end{bmatrix}^{\mathrm{T}}), \quad b_3^- = \exp_{p_3}(-\frac{\pi}{8\sqrt{2}}\begin{bmatrix}-1 & 1 & 0\end{bmatrix}^{\mathrm{T}}).$$

The remaining control points b_1^- and b_2^- are given by the $\mathcal{C}^1$ conditions (4). The corresponding curve $\mathbf{B}(t)$, $t \in [0, 3]$, is shown in **Figure 9** in dashed black. When computing a minimal MSA curve that interpolates $\mathbf{B}(t)$ at the points p_i, $i = 0, \dots, 3$, the acceleration is not that much reduced, see the blue curve in **Figure 9A**.

For fitting, we consider different values of λ and the same parameters as for the last example. The optimized curve fits the data points closer and closer as λ grows, and the limit $\lambda \to \infty$ yields the interpolation case. On the other hand smaller values of λ yield less fitting, but also a smaller value of the mean squared acceleration. In the limit case, i.e., $\lambda = 0$, the curve (more precisely the control points of the Bézier curve) just follows the gradient flow to a geodesic.

The results are collected in **Figure 9**. In **Figure 9A**, the original curve (dashed) is shown together with the solution of the interpolating model (solid blue) and the gradient flow result, i.e., the solution of the fitting model (solid black) with $\lambda = 0$. **Figure 9B** illustrates the effect of λ even further with a continuous variation of λ from a large value of $\lambda = 10$ to a small value of $\lambda = 0.01$. The image is generated by sampling this range with 1000 equidistant values colored in the colormap `viridis`. Furthermore, the control points are also shown in the same color.

Several corresponding functional values, see **Table 1**, further illustrate that with smaller values of λ the discretized MSA also reduces more and more to a geodesic. For $\lambda = 0$ there is no coupling to the data points and hence the algorithm does not

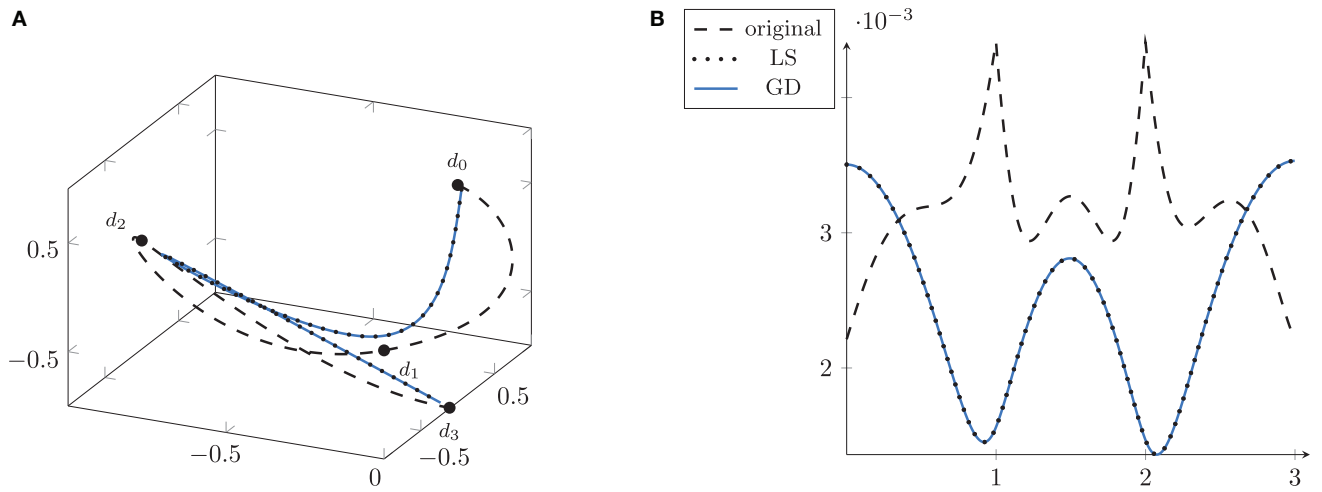

FIGURE 7 | The initial interpolating Bézier curve in $\mathbb{R}^3$ (dashed, **A**) with an MSA of 18.8828 is optimized by the linear system method (LS) from [37] (dotted) and by the proposed gradient descent (GD, solid blue). As expected, both curve coincide, with an MSA of 4.981218. The correspondance is further illustrated with their first order differences in **(B)**.

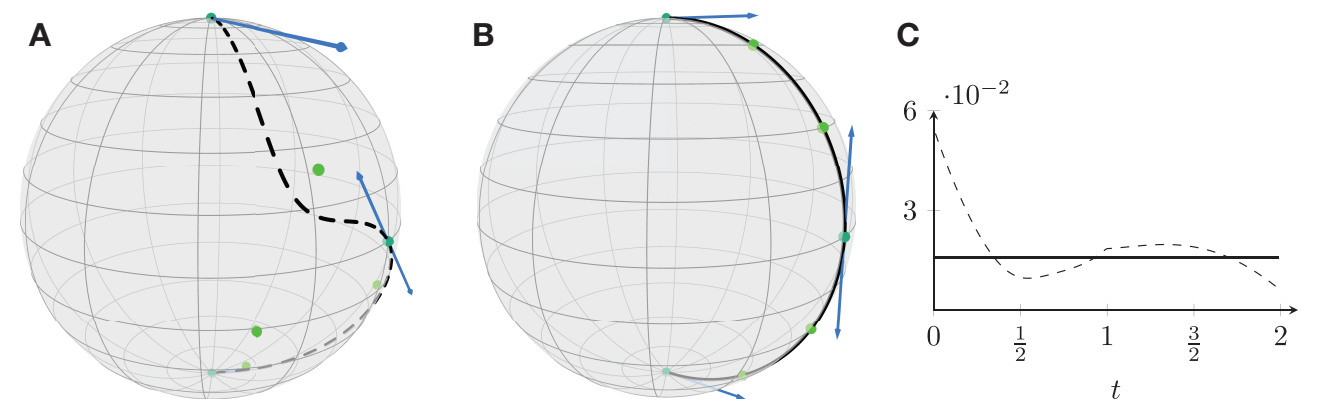

FIGURE 8 | The initial curve (dashed, **A**) results in a geodesic (solid, **B**) when minimizing the discrete acceleration. This can also be seen on the first order differences **(C)**.

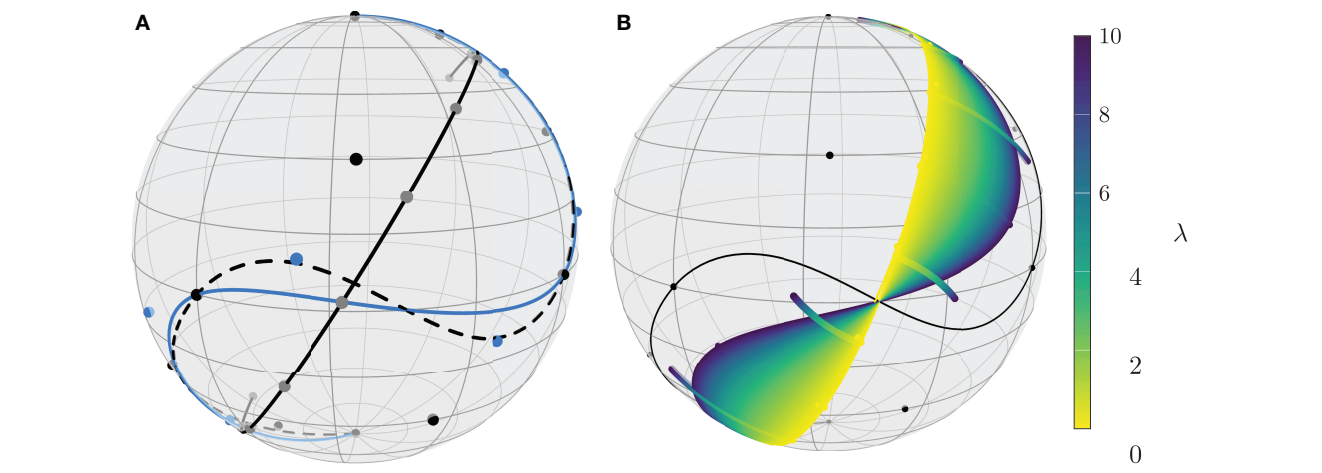

FIGURE 9 | The composite Bézier curves are composed of three segments. The initial curve (dashed, **A**) is optimized with the interpolating model (solid blue, **A**) as well as with the fitting model for a continuum of values of λ, from $\lambda = 10$ (violet, **B**) to $\lambda = 0.01$ (yellow, **B**). The limit case where $\lambda = 0$ yields an unconstrained geodesic (solid black, **A**).

restrict the position of the geodesic. In other words, any choice for the control points that yields a geodesic, is a solution. Note that the gradient flow still chooses a reasonably near geodesic to the initial data (**Figure 9A**, solid black).

Comparison with the tangential solution. In the Euclidean space, the method introduced in Arnould et al. [37] and Gousenbourger et al. [38] and the gradient descent proposed here yield the same curve, i.e., the natural cubic spline. On Riemannian manifolds, however, all approaches approximate the optimal solution. Indeed, their method provides a solution by working in different tangent spaces, and ours minimizes a discretization of the objective functional.

In this example we take the same data points d_i as in (25) now interpreted as points on $\mathcal{M} = \mathbb{S}^2$. Note that the control points constructed in (26) still fit to the sphere since each $b_i^{\pm}$ is built with a vector in $T_{p_i}\mathcal{M}$ and using the corresponding exponential map. The result is shown in **Figure 10A**. The norm of the first order differences is given in **Figure 10B**. The initial curve (dashed black) has an objective value of 10.9103. The tangent version (dotted) reduces this value to 7.3293. The proposed method (solid blue) yields a value of 2.7908, regardless of whether the starting curve is the initial one, or the one already computed by Arnould et al. [37]. Note that, when $p_3 = [0, 0, -1]^{\mathrm{T}}$, the tangent version is even not able to compute any result, since the construction is performed in the tangent space of p_0 and since $\log_{p_0} p_3$ is not defined.

TABLE 1 | Functional values for the three-segment composite Bézier curve on the sphere and different values of λ.

λ	$\tilde{A}$(b)
original	10.6122
∞	4.1339
10	1.6592
1	0.0733
0.1	0.0010
0.01	1.0814e-5
0.001	1.6240e-7
0	3.5988e-9

Note that the case $\lambda = \infty$ corresponds to interpolation.

5.3. An Example of Orientations

Finally we compare our method with the blended splines introduced in Gousenbourger et al. [38] for orientations, i.e., data given on SO(3). Let

$$R_{\mathrm{xy}}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_{\mathrm{xz}}(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix},$$
$$R_{\mathrm{yz}}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$$

denote the rotation matrices in the $x-y$, $x-z$, and $y-z$ planes, respectively. We introduce the three data points

$$d_0 = R_{\mathrm{xy}}\left(\frac{4\pi}{9}\right) R_{yz}\left(-\frac{\pi}{2}\right),$$
$$d_1 = R_{\mathrm{xz}}\left(-\frac{\pi}{8}\right) R_{\mathrm{xy}}\left(\frac{\pi}{18}\right) R_{yz}\left(-\frac{\pi}{2}\right),$$
$$d_2 = R_{\mathrm{xy}}\left(\frac{5\pi}{9}\right) R_{yz}\left(-\frac{\pi}{2}\right).$$

These data points are shown in the first line of **Figure 11** in cyan.

We set $\lambda = 10$ and discretize (20) with $N = 401$ equispaced points. This sampling is also used to generate the first order finite

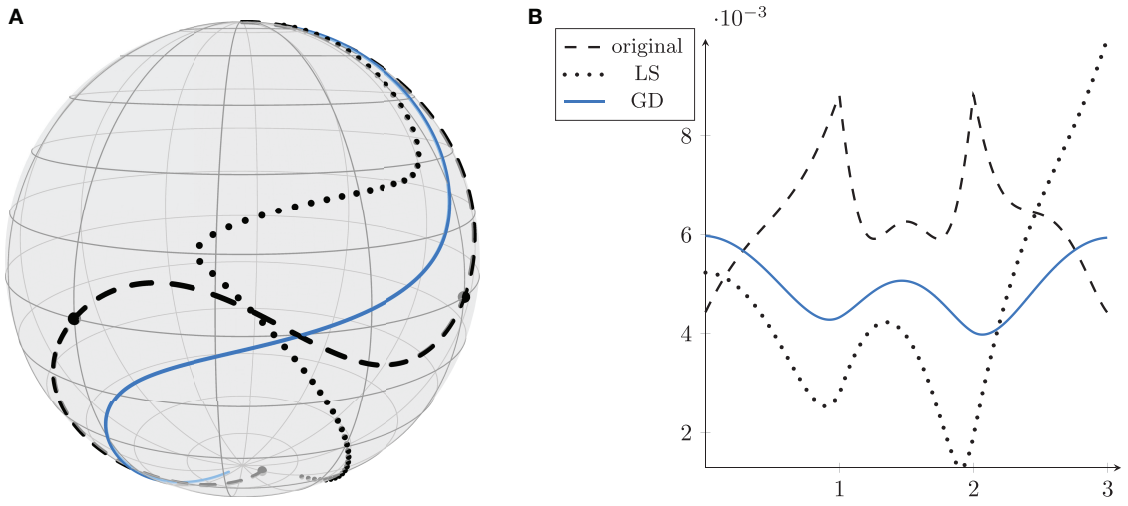

FIGURE 10 | The initial composite Bézier curve on $\mathbb{S}^2$ (dashed, **A**) has an MSA of 10.9103. The curve obtained with the linear system (LS, dotted) of [37] has an MSA of 7.3293. The solution returned by our proposed method (GD, solid blue) outperforms them all with an MSA of 2.7908. This is further illustrated by the first order derivative approximated by first order differences in **(B)**.

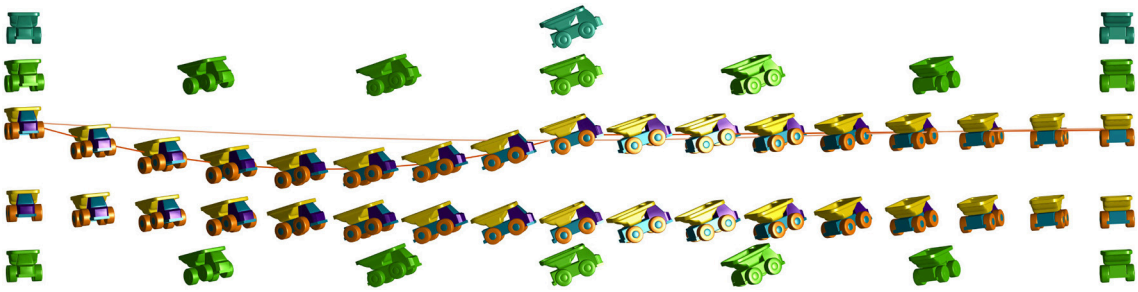

FIGURE 11 | From top row to bottom row: (1) the initial data points (cyan), (2) the control points computed by the blended Bézier approach from Gousenbourger et al. [38], (3) 17 points on the corresponding curve, where the height represents the absolute first order difference, and the comparing curve is the first order differences from our model, (4) 17 points along the resulting curve from gradient descent, and (5) the resulting control points of the curve in (4).

differences. The parameters of the gradient descent algorithm are set to the same values as for the examples on the sphere.

We perform the blended spline fitting with two segments and cubic splines. The resulting control points are shown in the second row of **Figure 11**, in green. The objective value is $\tilde{A}(\mathbf{b}) = 0.6464$. We improve this solution with the minimization problem (22) on the product manifold $\mathrm{SO}(3)^6$. We obtain the control points shown in the last line of **Figure 11** and an objective value of $\tilde{A}(\hat{\mathbf{b}}) = 0.2909$ for the resulting minimizer $\hat{\mathbf{b}}$.

We further compare both curves by looking at their absolute first order differences. In the third line of **Figure 11**, we display 17 orientations along the initial curve with its first order difference as height. The line without samples is the absolute first order differences of the minimizer $\hat{\mathbf{b}}$, scaled to the same magnitude. The line is straightened especially for the first Bézier segment. Nevertheless, the line is still bent a little bit and hence not a geodesic. This can be seen in the fourth line which represents 17 samples of the minimizing curve compared to its control points in the last line, which are drawn on a straight line.

6. CONCLUSION AND FUTURE WORK

In this paper, we introduced a method to solve the curve fitting problem to data points on a manifold, using composite Bézier curves. We approximate the mean squared acceleration of the curve by a suitable second order difference and a trapezoidal rule, and derive the corresponding gradient with respect to its control points. The gradient is computed in closed form by exploiting the recursive structure of the De Casteljau algorithm. Therefore, we obtain a formula that reduces to a concatenation of adjoint Jacobi fields.

The evaluation of Jacobi fields is the only additional requirement compared to previous methods, which are solely evaluating exponential and logarithmic maps. For these, closed forms are available on symmetric manifolds.

On the Euclidean space our solution reduces to the natural smoothing spline, the unique acceleration minimizing polynomial curve. The numerical experiments further confirm that the method presented in this paper outperforms the tangent space(s)-based approaches with respect to the functional value.

It is still an open question whether there exists a second order absolute finite difference on a manifold, that is jointly convex. Then convergence would follow by standard arguments of the gradient descent. For such a model, another interesting point for future work is to find out whether only an approximate evaluation of the Jacobi fields suffices for convergence. This would mean that, on manifolds where the Jacobi field can only be evaluated approximately by solving an ODE, the presented approach would still converge.

AUTHOR CONTRIBUTIONS

Both authors listed have contributed equally to the content of this work and approved the paper for publication.

ACKNOWLEDGMENTS

This work was supported by the following fundings: RB gratefully acknowledges support by DFG grant BE 5888/2–1; P-YG gratefully acknowledges support by the Fonds de la Recherche Scientifique – FNRS and the Fonds Wetenschappelijk Onderzoek – Vlaanderen under EOS Project no 30468160, and Communauté française de Belgique - Actions de Recherche Concertées (contract ARC 14/19-060). **Figure 9A** is based on an idea of Rozan I. Rosandi.

REFERENCES

1. Pyta L, Abel D. Interpolatory Galerkin models for Navier-Stokes-equations. *IFAC-PapersOnLine* (2016) **49**:204–9. doi: 10.1016/j.ifacol.2016.07.442
2. Gousenbourger PY, Massart E, Musolas A, Absil PA, Jacques L, Hendrickx JM, et al. Piecewise-Bézier C1 smoothing on manifolds with application to wind field estimation. In: *ESANN2017*. Bruges: Springer (2017). Available online at: https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2017-77.pdf
3. Sander O. Geodesic finite elements for cosserat rods. *Int J Numer Methods Eng.* (2010) **82**:1645–70. doi: 10.1002/nme.2814
4. Park J. Interpolation and tracking of rigid body orientations. In: *ICCAS*, Gyeonggi-do (2010). p. 668–73. doi: 10.1109/ICCAS.2010.5670237

5. Su J, Kurtek S, Klassen E, Srivastava A. Statistical analysis of trajectories on Riemannian manifolds: bird migration, Hurricane tracking and video surveillance. *Ann Appl Stat*. (2014) **8**:530–52. doi: 10.1214/13-AOAS701
6. Dyn N. Linear and nonlinear subdivision schemes in geometric modeling. In: Cucker F, Pinkus A, Todd MJ, editors. *FoCum. Vol. 363 of London Math. Soc. Lecture Note Ser*. Hong Kong: Cambridge University Press (2009). p. 68–92. doi: 10.1017/CBO9781139107068.004
7. Wallner J, Nava Yazdani E, Grohs P. Smoothness properties of Lie group subdivision schemes. *Multiscale Model Simulat*. (2007) **6**:493–505. doi: 10.1137/060668353
8. Shingel T. Interpolation in special orthogonal groups. *IMA J Numer Anal*. (2008) **29**:731–45. doi: 10.1093/imanum/drn033
9. Green PJ, Silverman BW. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. CRC Press (1993). Available online at: https://www.crcpress.com/Nonparametric-Regression-and-Generalized-Linear-Models-A-roughness-penalty/Green-Silverman/p/book/9780412300400
10. Absil PA, Mahony R, Sepulchre R. *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press (2008). Available online at: https://press.princeton.edu/absil
11. Boumal N, Mishra B, Absil PA, Sepulchre R. Manopt, a matlab toolbox for optimization on manifolds. *J Mach Learn Res*. (2014) **15**:1455–9. Available online at: http://jmlr.org/papers/v15/boumal14a.html
12. Bergmann R. MVIRT, a toolbox for manifold-valued image registration. In: *IEEE International Conference on Image Processing, IEEE ICIP 2017, Beijing*. (2017). Available online at: https://ronnybergmann.net/mvirt/
13. Zhou X, Yang C, Zhao H, Yu W. Low-rank modeling and its applications in image analysis. *ACM Comput Surveys* (2015) **47**:36. doi: 10.1145/2674559
14. Cunningham JP, Ghahramani Z. Linear dimensionality reduction: survey, insights, and generalizations. *J Mach Learn Res*. (2015) **16**:2859–900. Available online at: http://jmlr.org/papers/v16/cunningham15a.html
15. Sun J, Qu Q, Wright J. A geometric analysis of phase retrieval. *Found Comput Math*. (2017) **8**:1131–98. doi: 10.1007/s10208-017-9365-9
16. Yu X, Shen JC, Zhang J, Letaief KB. Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems. *IEEE J Selected Top Signal Process*. (2016) **10**:485–500. doi: 10.1109/JSTSP.2016.2523903
17. Strekalovskiy E, Cremers D. Total variation for cyclic structures: convex relaxation and efficient minimization. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2011). p. 1905–11.
18. Strekalovskiy E, Cremers D. Total cyclic variation and generalizations. *J Mat Imaging Vis*. (2013) **47**:258–77. doi: 10.1007/s10851-012-0396-1
19. Lellmann J, Strekalovskiy E, Koetter S, Cremers D. Total variation regularization for functions with values in a manifold. In: *IEEE International Conference on Computer Vision*. Sydney, NSW (2013). p. 2944–51. doi: 10.1109/ICCV.2013.366
20. Weinmann A, Demaret L, Storath M. Total variation regularization for manifold-valued data. *SIAM J Imaging Sci*. (2014) **7**:2226–57. doi: 10.1137/130951075
21. Bergmann R, Laus F, Steidl G, Weinmann A. Second order differences of cyclic data and applications in variational denoising. *SIAM J Imaging Sci*. (2014) **7**:2916–53. doi: 10.1137/140969993
22. Bačák M, Bergmann R, Steidl G, Weinmann A. A second order nonsmooth variational model for restoring manifold-valued images. *SIAM J Comput*. (2016) **38**:567–97. doi: 10.1137/15M101988X
23. Bergmann R, Fitschen JH, Persch J, Steidl G. Priors with coupled first and second order differences for manifold-valued image processing. *J Math Imaging Vis*. (2017) **60**:1459–81. doi: 10.1007/s10851-018-0840-y
24. Bredies K, Holler M, Storath M, Weinmann A. Total generalized variation for manifold-valued data. *SIAM J Imaging Sci*. (2018) **11**:1785–48. doi: 10.1137/17M1147597
25. Bergmann R, Persch J, Steidl G. A parallel Douglas–Rachford algorithm for restoring images with values in symmetric Hadamard manifolds. *SIAM J Imaging Sci*. (2016) **9**:901–37. doi: 10.1137/15M1052858
26. Grohs P, Sprecher M. Total variation regularization on Riemannian manifolds by iteratively reweighted minimization. *Inform Inference* (2016) **5**:353–78. doi: 10.1093/imaiai/iaw011
27. Bergmann R, Chan RH, Hielscher R, Persch J, Steidl G. Restoration of manifold-valued images by half-quadratic minimization. *Inverse Prob Imaging* (2016) **10**:281–304. doi: 10.3934/ipi.2016001
28. Samir C, Absil PA, Srivastava A, Klassen E. A gradient-descent method for curve fitting on Riemannian manifolds. *Found Comput Math*. (2012) **12**:49–73. doi: 10.1007/s10208-011-9091-7
29. Su J, Dryden IL, Klassen E, Le H, Srivastava A. Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds. *Image Vis Comput*. (2012) **30**:428–42. doi: 10.1016/j.imavis.2011.09.006
30. Boumal N, Absil PA. A discrete regression method on manifolds and its application to data on SO(n). In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*. Vol. 18. Milano (2011). p. 2284–89. doi: 10.3182/20110828-6-IT-1002.00542
31. Kim KR, Dryden IL, Le H. Smoothing splines on Riemannian manifolds, with applications to 3D shape space. (2018). *arXiv[Preprint]*:1801.04978.
32. Fletcher PT. Geodesic regression and the theory of least squares on Riemannian manifolds. *Int J Comput Vis*. (2013) **105**:171–85. doi: 10.1007/s11263-012-0591-y
33. Rentmeesters Q. 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*. Orlando, FL (2011). p. 7141–6. doi: 10.1109/CDC.2011.6161280
34. Farin G. *Curves and Surfaces for CAGD*. Morgan Kaufmann (2002). Available online at: https://www.elsevier.com/books/curves-and-surfaces-for-cagd/farin/978-1-55860-737-8
35. Absil PA, Gousenbourger PY, Striewski P, Wirth B. Differentiable Piecewise-Bézier Surfaces on Riemannian Manifolds. *SIAM J Imaging Sci*. (2016) **9**:1788–828. doi: 10.1137/16M1057978
36. Popiel T, Noakes L. Bézier curves and C^2 interpolation in Riemannian manifolds. *J Approx Theory* (2007) **148**:111–27. doi: 10.1016/j.jat.2007.03.002
37. Arnould A, Gousenbourger PY, Samir C, Absil PA, Canis M. Fitting smooth paths on Riemannian manifolds: endometrial surface reconstruction and preoperative MRI-based navigation. In: Nielsen F, Barbaresco F, editors. *GSI2015*. Springer International Publishing (2015). p. 491–8.
38. Gousenbourger PY, Massart E, Absil PA. Data fitting on manifolds with composite Bézier-like curves and blended cubic splines. *J Math Imaging Vis*. (2018). doi: 10.1007/s10851-018-0865-2
39. O'Neill B. *Elementary Differential Geometry*. London: Academic Press Inc. (1966).
40. do Carmo MP. *Riemannian Geometry*. Vol. 115. Basel: Birkhäuser; 1992. Tranl. by F. Flatherty.
41. Karcher H. Riemannian center of mass and mollifier smoothing. *Commun Pure Appl Math*. (1977) **30**:509–41. doi: 10.1002/cpa.3160300502
42. Boumal N. Interpolation and regression of rotation matrices. In: Nielsen F, Barbaresco F, editors. *Geometric Science of Information*. Berlin; Heidelberg: Springer. (2013). p. 345–52.
43. Persch J. *Optimization Methods in Manifold-Valued Image Processing [Dissertation]*. Technische Universität Kaiserslautern (2018).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.