

Fast method to fit a \mathcal{C}^1 piecewise-Bézier function to manifold-valued data points: how suboptimal is the curve obtained on the sphere \mathbb{S}^2 ?

Pierre-Yves Gousenbourger, Laurent Jacques, P.-A. Absil *

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

Abstract We propose an analysis of the quality of the fitting method proposed in [7]. This method fits smooth paths to manifold-valued data points using \mathcal{C}^1 piecewise-Bézier functions. This method is based on the principle of minimizing an objective function composed of a data-attachment term and a regularization term chosen as the mean squared acceleration of the path. However, the method strikes a tradeoff between speed and accuracy by following a strategy that is guaranteed to yield the optimal curve only when the manifold is linear. In this paper, we focus on the sphere \mathbb{S}^2 . We compare the quality of the path returned by the algorithms from [7] with the path obtained by minimizing, over the same search space of \mathcal{C}^1 piecewise-Bézier curves, a finite-difference approximation of the objective function by means of a derivative-free manifold-based optimization method.

Keywords: Path fitting on Riemannian manifolds, Bézier functions, optimization on manifolds.

1 Introduction

We consider the problem of fitting an univariate \mathcal{C}^1 piecewise-Bézier curve to manifold-valued data points. This problem is motivated by several applications in engineering and the sciences, such as projection-based model order reduction of dynamical systems that depend on one parameter [10]. In that case, the data points are projectors from the full state space to the reduced state space and hence belong to a Grassmann manifold. In a recent paper, Gousenbourger *et al.* [7] illustrated the benefits of this approach by estimating wind fields: the task required to fit a curve to a set of data points belonging to the manifold of $p \times p$ positive semidefinite (PSD) matrices of rank r . We also mention the case of image denoising, as in Bergmann *et al.* [3], where one seeks a two-parameter function fitting an image with manifold-valued pixels, or the blood vessels tracking in the eyes in Sanguinetti *et al.* [12] as an application to the sub-Riemannian manifold $SE(2)$.

* Corresponding author: pierre-yves.gousenbourger@uclouvain.be

Fitting and interpolation on manifolds has been an active research topic in the past few years. For instance Samir *et al.* [11] proposed a fitting method where the search space is infinite-dimensional. In that paper, the fitting curve \mathfrak{B} is discretized with a small stepsize and the objective function is minimized with a manifold-valued gradient descent. An application in image processing can be found in Su *et al.* [13]. In Absil *et al.* [1] (interpolation) or more recently in Gousenbourger *et al.* [7] (fitting), the search space is restricted to a finite dimensional space of C^1 piecewise-Bézier functions. We also mention Machado *et al.* [8] for the specific case of the sphere.

The method proposed in [7] seeks a C^1 piecewise-Bézier curve as in [2]. It also considers a smoothing objective function—a roughness penalty and a data-fitting term—as in [11]. This approach has several advantages. With respect to [2], interpolation is replaced by smoothing, which is more apt for noisy data. Compared to [11], (i) it reduces the space complexity (instead of being discretized, the solution curve is represented by only a few Bézier control points on the manifold) and (ii) it provides a very simple algorithm that only requires two objects on the manifold: the Riemannian exponential and the Riemannian logarithm. However, the proposed approach tends to be suboptimal for two reasons. First, the search space is restricted to C^1 piecewise-Bézier curves; and second, the proposed computational method ensures optimality (within the restricted search space) only if the manifold is flat.

The study of this second drawback is the subject of this paper: in particular, we aim to evaluate the quality of the fitting curve obtained with the method developed in [7] compared to a more accurate solution obtained with a more general (but also slower) optimization tool (like, for instance, Manopt [5]).

The paper is organized as follows. We first recall some generalities on Bézier curves and introduce the composite Bézier curve \mathfrak{B} we would like to fit to data points (Section 2). In Section 3, we summarize the method from [7] and then introduce a more accurate (but also less efficient) method based on a discretization. We also look for an acceptable discretization stepsize. Finally, we present results on the sphere \mathbb{S}^2 in Section 4

2 Notations and framework

We consider the case in which the data points $\{d_0, \dots, d_n\} \subset \mathcal{M}$ take values on a manifold \mathcal{M} and are associated with measurement parameters $t_0 \leq t_1 \leq \dots \leq t_n$. For simplicity, we will let $t_i = i$, $i = 0, \dots, n$. We seek a composite Bézier curve $\mathfrak{B} : \mathbb{R} \rightarrow \mathcal{M}$ such that $\mathfrak{B}(t_i) \simeq d_i$, $i = 0, \dots, n$. We note $T_a\mathcal{M}$ the (Euclidean) tangent space to \mathcal{M} at $a \in \mathcal{M}$; $T\mathcal{M} = \cup_a T_a\mathcal{M}$ the tangent bundle to \mathcal{M} ; $\langle \cdot, \cdot \rangle_a$, the inner product in the tangent space at a and from which we deduce the norm of $v \in T_a\mathcal{M}$, $\|v\|_{\mathcal{M}} = \langle v, v \rangle_a$; $\exp_a(\cdot) : T_a\mathcal{M} \rightarrow \mathcal{M} : v \mapsto b = \exp_a(v)$, the Riemannian exponential; $\log_a(\cdot) : \mathcal{M} \rightarrow T_a\mathcal{M} : b \mapsto v = \log_a(b)$, the Riemannian logarithm which can be viewed as the inverse Riemannian exponential. We also introduce the notation $\gamma_{a,b}(t)$ for the shortest geodesic between $a = \gamma_{a,b}(0)$ and $b = \gamma_{a,b}(1)$. We assume throughout that we can compute these objects.

2.1 Preliminaries on Bézier curves

We first consider the trivial case where $\mathcal{M} = \mathbb{R}^r$ to define the Bézier curve. A *Bézier curve of degree* $K \in \mathbb{N}$ is a function β parametrized by $K + 1$ *control points* $\{b_0, \dots, b_K\} \subset \mathbb{R}^r$ taking the form

$$\beta_K(\cdot; b_0, \dots, b_K) : [0, 1] \rightarrow \mathbb{R}^r, t \mapsto \sum_{j=0}^K b_j B_{jK}(t),$$

where $B_{jK}(t) = \binom{K}{j} t^j (1-t)^{K-j}$ are the Bernstein basis polynomials (also called binomial functions) [6]. The first control point and the last one are interpolated by construction while the position of the other control points models the shape of the curve. More specifically, the quadratic and cubic Bézier curves are respectively

$$\beta_2(t; b_0, b_1, b_2) = b_0(1-t)^2 + 2b_1(1-t)t + b_2t^2 \quad (1)$$

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

One well-known way to generalize Bézier curves to a Riemannian manifold \mathcal{M} is via the De Casteljau algorithm. This algorithm, generalized to manifolds by Popiel and Noakes [9, §2], only requires the Riemannian exponential and logarithm and conserves the interpolation property of the first and last control points.

2.2 Composite Bézier function on manifolds

We now consider a general manifold \mathcal{M} . As illustrated in Figure 1, the composite Bézier function $\mathfrak{B} \in \mathcal{M}$ is a \mathcal{C}^1 composition of n Bézier curves, *i.e.*,

$$\mathfrak{B} : [0, n] \rightarrow \mathcal{M}, t \mapsto \beta^i(t-i) \text{ on } [i, i+1], \quad i = 0, \dots, n-1,$$

where β^i defines a piece of \mathfrak{B} associated to the endpoints $\{p_i, p_{i+1}\} \subset \mathcal{M}$. The control points of the $(i-1)^{\text{th}}$ and i^{th} piece of \mathfrak{B} defined on the left and right of p_i are noted $\{b_i^-, b_i^+\} \subset \mathcal{M}$, $i = 1, \dots, n-1$. The first and last segments of \mathfrak{B} are quadratic Bézier curves respectively noted $\beta^0(t) = \beta_2(t; p_0, b_1^-, p_1)$ and $\beta^{n-1}(t) = \beta_2(t; p_{n-1}, b_{n-1}^+, p_n)$. All the other segments are cubic and denoted by $\beta^i(t) = \beta_3(t; p_i, b_i^+, b_{i+1}^-, p_{i+1})$. Note the use of the superscript to refer to the i^{th} segment of \mathfrak{B} while the subscript refers to the degree of the Bézier curve.

The continuity of \mathfrak{B} is trivial as $\mathfrak{B}(i) = \beta^i(i) = \beta^{i-1}(i) = p_i$. Differentiability is ensured by taking $p_1 = \text{av}[(b_1^-, b_1^+), (\frac{2}{5}, \frac{3}{5})]$, $p_i = \text{av}[(b_i^-, b_i^+), (\frac{1}{2}, \frac{1}{2})]$ ($i = 2, \dots, n-2$) and $p_{n-1} = \text{av}[(b_{n-1}^-, b_{n-1}^+), (\frac{3}{5}, \frac{2}{5})]$, where $\text{av}[(x, y), (1-\alpha, \alpha)] = \exp_x(\alpha \log_x(y))$ stands for the convex combination of $x, y \in \mathcal{M}$ with weight $\alpha \in [0, 1]$. A proof of these properties can be found in [1].

As stated in the introduction, we would ideally like \mathfrak{B} to minimize its mean square acceleration and its fidelity to data points. Specifically,

$$\min_{p_i, b_i^+, b_i^-} f(p_i, b_i^+, b_i^-) = \underbrace{\min_{p_i, b_i^+, b_i^-} \sum_{i=0}^{n-1} \int_0^1 \|\ddot{\beta}^i(t)\|_{\mathcal{M}}^2 dt}_{\text{"mean square acceleration"}} + \lambda \underbrace{\sum_{i=0}^n d^2(p_i, d_i)}_{\text{"fidelity"}}, \quad (3)$$

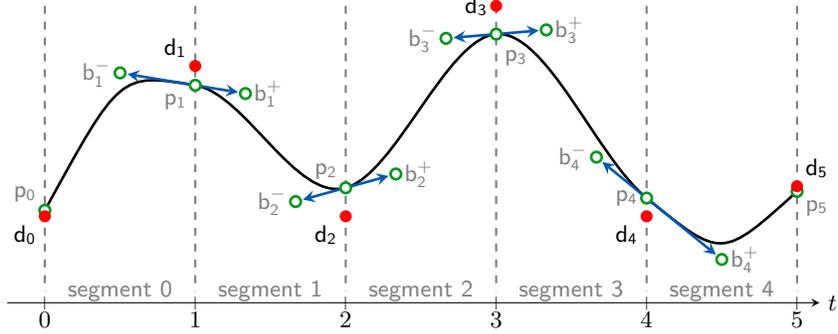


Figure 1: Schematic representation of the composite Bézier function $\mathfrak{B}(t)$: the data points d_i are represented in red; the circled green ones are control points. The first and last Bézier segments are quadratic Bézier functions while all the other segments are cubic Bézier functions.

where $\ddot{g}(t)$ stands for the temporal covariant second derivative of $g(t)$, under continuity and differentiability constraints. The parameter $\lambda > 0$ adjusts the balance between data fidelity and the “smoothness” of \mathfrak{B} . This balance tends to the interpolation problem from [2] when $\lambda \rightarrow \infty$.

3 Methods

In this section, we first summarize the method from [7] which is a generalization of optimality conditions holding only when $\mathcal{M} = \mathbb{R}^r$. This generalization holds for any manifold \mathcal{M} if it is possible to compute the exponential map and the logarithm map. In a second time, we introduce a version $\tilde{f}_{\Delta\tau}$ of the objective f (equation (3)) obtained by discretizing the time domain of the mean square acceleration term with a step size $\Delta\tau$. We determine experimentally $\Delta\tau$ for which $\tilde{f}_{\Delta\tau}$ is a sufficiently good approximation of f , *i.e.*, the relative error between $\tilde{f}_{\Delta\tau}$ and f is small. Then, in Section 4, we will compare the solution from [7] to the minimizer of $\tilde{f}_{\Delta\tau}$.

3.1 Summary of the optimality conditions from [7]

In [7], the problem (3) is not directly addressed on a manifold \mathcal{M} . The (suboptimal) solution is obtained in two steps.

Step 1. The problem is considered on $\mathcal{M} = \mathbb{R}^r$ where $d(\cdot, \cdot)$ and $\|\cdot\|_{\mathcal{M}}$ are the classical Euclidean distance and norm, respectively. Hence (3) is a quadratic function in the $2n$ variables p_0 , $(b_i^-, b_i^+)_{i=1}^{n-1}$ and p_n . Therefore, its optimality conditions take the form of a linear system $(A_0 + \lambda A_1)\mathbf{x} = \lambda C\mathbf{d}$, where $A_0, A_1 \in \mathbb{R}^{2n \times 2n}$ and $C \in \mathbb{R}^{2n \times n+1}$ are matrices of coefficients, where $\mathbf{x} = [x_0, x_1, \dots, x_{2n-1}]^T := [p_0, b_1^-, b_1^+, \dots, b_{n-1}^-, b_{n-1}^+, p_n]^T \in \mathbb{R}^{2n \times r}$ contains the $2n$

optimization variables, and where $\mathbf{d} := [d_0, \dots, d_n]^T \in \mathbb{R}^{n+1 \times r}$ contains the data points. The solution reads $\mathbf{x} = Q(\lambda)\mathbf{d}$, or

$$x_j = \sum_{l=0}^n q_{jl}(\lambda)d_l, \quad (4)$$

with $Q(\lambda) \in \mathbb{R}^{2n \times n+1}$, a matrix of coefficients depending on λ .

Step 2. Because (3) is invariant to translation on \mathbb{R}^r , the conditions (4) can be generalized to any Riemannian manifold. Indeed, $x_j - d_j^* = \sum_{l=0}^n q_{jl}(\lambda)(d_l - d_j^*)$, by translation with respect to a reference point d_j^* . The generalization arises by interpreting the Euclidean difference as a logarithm map on a general manifold \mathcal{M} . Thus, a simple and natural way to generalize (4) to \mathcal{M} is

$$x_j = \exp_{d_j^*} \left(\sum_{l=0}^n q_{jl}(\lambda) \log_{d_j^*}(d_l) \right). \quad (5)$$

By default, $d_j^* := d_i$ when x_j is one of the control points b_i^- , b_i^+ or p_i .

Finally, the curve \mathfrak{B} is reconstructed using the De Casteljau algorithm (as mentioned in Section 2).

3.2 Discretization of the mean square acceleration on manifolds

In comparison to Section 3.1, we here solve (3) directly on an arbitrary manifold \mathcal{M} . However, there is no simple expression of the Bézier curves β^i on \mathcal{M} , which means that it is not possible to express its mean squared acceleration in general. To overcome this difficulty, we replace f by a version $\tilde{f}_{\Delta\tau}$ where the acceleration of the curves is approached by a Riemannian second order finite difference (generalized with the log map from the Euclidean finite differences as in [4]), and the integration is replaced by a classical trapezoidal rule. The new objective function $\tilde{f}_{\Delta\tau}(p_i, b_i^+, b_i^-)$ now reads

$$\sum_{k=1}^{M-1} \Delta\tau \left\| \frac{\log_{\mathfrak{B}(t_k)}(\mathfrak{B}(t_{k-1})) + \log_{\mathfrak{B}(t_k)}(\mathfrak{B}(t_{k+1}))}{\Delta\tau^2} \right\|_{\mathcal{M}}^2 + \lambda \sum_{i=0}^n d^2(p_i, d_i), \quad (6)$$

where $\Delta\tau = \frac{n}{M}$. As there is also no general expression of the Riemannian gradient of $\tilde{f}_{\Delta\tau}$ with respect to p_0 , $(b_i^-, b_i^+)_{i=1}^{n-1}$ and p_n , we solve this problem with a Riemannian derivative-free optimization method, like the Particle Swarm Optimization algorithm provided in Manopt [5].

As there is no exact solution of (3) on a general Riemannian manifold, there is also no way to determine with precision the stepsize $\Delta\tau$ for which $\tilde{f}_{\Delta\tau}$ is close enough to f on \mathcal{M} . To overcome this, we determine an acceptable $\Delta\tau$ on the Euclidean space and then use this stepsize to optimize (6) on \mathcal{M} . This behavior is illustrated in Figure 2. We can see that a stepsize of $\Delta\tau \simeq 10^{-2}$ is already acceptable on the Euclidean space for $\tilde{f}_{\Delta\tau}$ to approach f with a relative error of less than 1%.

We will now use this stepsize to compare x^* , the solution (5) from Section 3.1, to \tilde{x}^* , the solution obtained with (6).

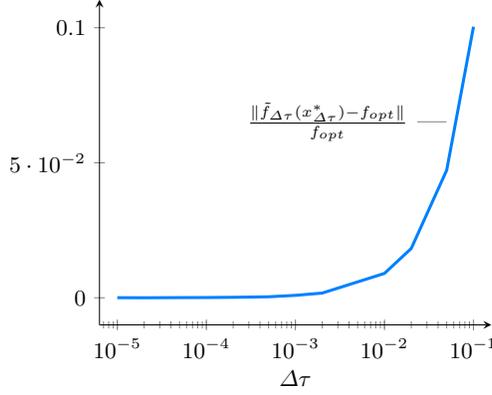


Figure 2: On the Euclidean space, the continuous objective function (3) is approached by its discretized version (6) when $\Delta\tau$ tends to be small. A stepsize of $\Delta\tau = 10^{-2}$ already leads to a relative error of less than 1% on a random set of data points.

4 Results

In this section, we evaluate the quality of the method from [7] (Section 3.1) on the sphere \mathbb{S}^2 . To do so, we compare its solution x^* with the solution \tilde{x}^* obtained by optimizing the discretized version of the objective function given in equation (6). This comparison is easily extendable to other manifolds provided that the log and exp map can be computed.

Data points and error evaluation. Consider the points $a = [0, 0, 1]^T$ and $b = [1, 0, 0]^T$ and the geodesic $\gamma_{a,b} : [0, 1] \rightarrow \mathcal{M} : t \mapsto \gamma_{a,b}(t)$. We construct $S = 20$ sets (indexed by m) of $n \in \{3, \dots, 10\}$ data points $(\hat{d}_i^m)_{i=1}^n$, $m = 1, \dots, S$ aligned and equispaced on the geodesic $\gamma_{a,b}(t)$ and then slightly disturbed with a noise η . Specifically,

$$\hat{d}_i^m = \frac{d_i^m + \eta}{\|d_i^m + \eta\|}, \quad i = 1, \dots, n, \quad m = 1, \dots, S,$$

such that $d_i^m = \gamma_{a,b}(\frac{i-1}{n-1})$, and $\eta \sim \mathcal{N}(0, (0.1)^2)$, as shown on Figure 3, left.

For each set m and each number of data points n , we compute $x_{m,n}^* \in \mathbb{S}^{2 \times 2n}$, the solution from [7] given by equation (5), and $\tilde{x}_{m,n}^* \in (\mathbb{S}^2)^{2n}$, the solution to the problem (6) with a discretization stepsize $\Delta\tau = 10^{-2}$. We evaluate the distance $\varepsilon_{m,n}$ of the objective value obtained with $x_{m,n}^*$ and $\tilde{x}_{m,n}^*$ in (6) as

$$\varepsilon_{m,n} = \frac{\tilde{f}_{\Delta\tau}(\tilde{x}_{m,n}^*) - \tilde{f}_{\Delta\tau}(x_{m,n}^*)}{\tilde{f}_{\Delta\tau}(\tilde{x}_{m,n}^*)}, \quad (7)$$

with $\Delta\tau = 10^{-4}$.

Note that two different stepsizes are used to evaluate the relative distance ε : a larger one ($\Delta\tau = 10^{-2}$) to compute \tilde{x}^* and another one ($\Delta\tau = 10^{-4}$) to

evaluate the quality of the solutions. We chose a larger stepsize in the minimization because solving (6) with a derivative free algorithm becomes less and less tractable when $\Delta\tau$ decreases. However, $\tilde{f}_{\Delta\tau}$ approaches the actual manifold-valued objective function (3) when $\Delta\tau$ is small. Thus, we used a finer stepsize to evaluate the quality of x^* in (7).

Results. On Figure 3 (right), we represent the mean $\mathcal{E}(n)$ and the standard deviation of the distances $(\varepsilon_{m,n})_{m=1}^S$, for each number n of data points. We can observe that the fast algorithm from [7] returns results close to the optimum in the case of this geodesic-like proof of concept, even if still slightly suboptimal (relative error of about 1% of the cost \tilde{f}). Indeed, this proof of concept might be too easy as data points are chosen close to a geodesic and Figure 3 (right) could be so good only in this case. However, finding a solution to the discretized problem with the particle-swarm optimization is less and less tractable for n growing and $\Delta\tau$ decreasing. This is why the main advantage of [7] is its efficiency to compute an acceptable solution to (3) in a very short computation time.

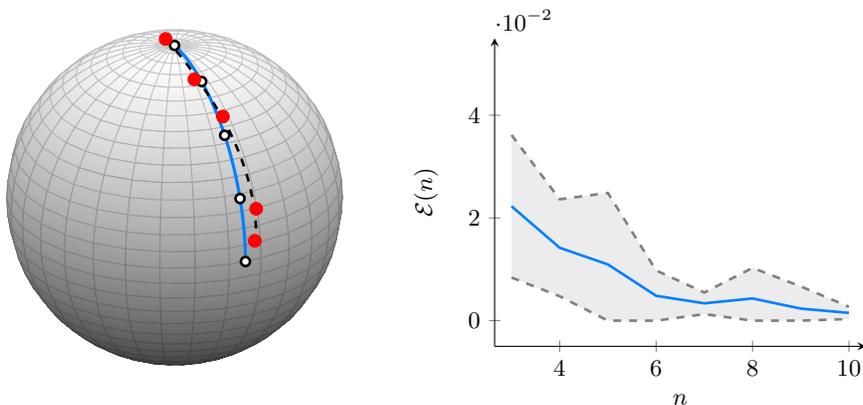


Figure 3: *Left* - the data points (red) are a noisy version of points (black circles) aligned on a geodesic (blue line). The Bézier curve computed via [7] based on the data points (red) is in dashed line. *Right* - the fast algorithm from [7] returns solutions close to optimum. The relative error $\mathcal{E}(n)$ is about 1% (solid) with a standard deviation (dashed) of 2%.

5 Future work

The sought goal in this paper was to evaluate the suboptimality of the fitting curve computed by the fast algorithm from [7]. We showed as a proof of concept that the method proposed in [7] approaches \tilde{x}^* with a very satisfactory small relative error of 1% of the cost \tilde{f} on the sphere \mathbb{S}^2 , when the data points lie close to a geodesic.

Different pieces of work can be considered for the future. For instance, it may be worth considering a more advanced configuration of the data points to evaluate better the limits of the method. Estimating a theoretical upper bound on $|f(x_{\text{opt}}) - f(x^*)|$, where x_{opt} is the actual (and not numerical) solution of (3) is also left for future work. Furthermore, using a derivative-free optimization tool appeared to be time-consuming: a gradient-based approach could be investigated, exploiting the iterative structure of the De Casteljau algorithm to approach the gradient of a general Bézier curve on \mathcal{M} .

References

- [1] Absil, P.A., Gouenbourger, P.Y., Striowski, P., Wirth, B.: Differentiable piecewise-Bézier surfaces on Riemannian manifolds. *SIAM Journal on Imaging Sciences* 9(4), pp. 1788–1828 (2016)
- [2] Arnould, A., Gouenbourger, P.Y., Samir, C., Absil, P.A., Canis, M.: Fitting Smooth Paths on Riemannian Manifolds : Endometrial Surface Reconstruction and Preoperative MRI-Based Navigation. In: F.Nielsen, F.Barbaresco (eds.) *GSi2015*. pp. 491–498. Springer International Publishing (2015)
- [3] Bergmann, R., Laus, F., Steidl, G., Weinmann, A.: Second order differences of cyclic data and applications in variational denoising. *SIAM Journal on Imaging Sciences* 7(4), pp. 2916–2953 (2014)
- [4] Boumal, N.: Interpolation and regression of rotation matrices. In: Nielsen, F. and Barbaresco, F. (ed.) *Geometric Science of Information*. vol. 8085. Springer Berlin Heidelberg (2013)
- [5] Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research* 15, pp. 1455–1459 (2014), <http://www.manopt.org>
- [6] Farin, G.: *Curves and Surfaces for CAGD*. Academic Press, fifth edition (2002)
- [7] Gouenbourger, P.Y., Massart, E., Musolas, A., Absil, P.A., Jacques, L., Hendrickx, J.M., Marzouk, Y.: Piecewise-Bézier C^1 smoothing on manifolds with application to wind field estimation. *ESANN2017*, to appear (2017)
- [8] Machado, L., Monteiro, M.T.T.: A numerical optimization approach to generate smoothing spherical splines. *Journal of Geometry and Physics* 111, pp. 71–81 (2017)
- [9] Popiel, T., Noakes, L.: Bézier curves and C^2 interpolation in Riemannian manifolds. *Journal on Approximation Theory* 148(2), pp. 111–127 (2007)
- [10] Pyta, L., Abel, D.: Interpolatory galerkin models for the navier-stokes-equations. *IFAC Papers online* 49(8), pp. 204 – 209 (2016)
- [11] Samir, C., Absil, P.A., Srivastava, A., Klassen, E.: A gradient-descent method for curve fitting on Riemannian manifolds. *Foundations of Computational Mathematics* 12, pp. 49–73 (2012)
- [12] Sanguinetti, G., Bekkers, E., Duits, R., Janssen, M.H.J., Mashtakov, A., Mirebeau, J.M.: Sub-riemannian fast marching in $SE(2)$. In: *Lecture Notes in Computer Science*. vol. 9423, pp. 366–374 (2015)
- [13] Su, J., Dryden, I., Klassen, E., Le, H., Srivastava, A.: Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds. *Image and Vision Computing* 30(67), pp. 428 – 442 (2012)