

# Piecewise-Bézier $C^1$ smoothing on manifolds with application to wind field estimation

Pierre-Yves Gousenbourger<sup>1</sup>, Estelle M. Massart<sup>1</sup>, Antoni Musolas<sup>2</sup>,  
P.-A. Absil<sup>1</sup>, Julien M. Hendrickx<sup>1</sup>, Laurent Jacques<sup>1</sup>, Youssef Marzouk<sup>2</sup>. \*

1- ICTEAM Institute, Université catholique de Louvain, Louvain-la-Neuve, Belgium

2- Dept. of Aeronautics and Astronautics, MIT, Cambridge, USA

**Abstract.** We propose an algorithm for fitting  $C^1$  piecewise-Bézier curves to (possibly corrupted) data points on manifolds. The curve is chosen as a compromise between proximity to data points and regularity. We apply our algorithm as an example to fit a curve to a set of low-rank covariance matrices, a task arising in wind field modeling. We show that our algorithm has denoising abilities for this application.

## 1 Introduction

This paper concerns univariate manifold-valued data approximation by means of  $C^1$  piecewise-Bézier curves.

This task is motivated, among other applications, by parametric model reduction problems [1] where the data points are projectors from the full state space to the reduced state space and hence belong to the Grassmann manifold.

We illustrate here the benefits of our algorithm by applying it to wind field estimation, which requires to fit a curve to a set of data points belonging to the manifold of  $p \times p$  positive semidefinite (PSD) matrices of rank  $r$ . The wind field estimation problem is motivated by applications of unmanned aerial vehicles (UAV). Safe and reliable navigation of UAVs requires consideration of the surrounding environment, in particular, the external wind conditions. The wind field around the UAV can be modelled as a Gaussian process characterized by a covariance matrix that is itself parameterized by external meteorological parameters (*e.g.*, the prevailing wind in the area of interest). For each prevailing wind, computationally expensive unsteady CFD simulations are run to estimate the corresponding covariance matrix. We propose here to run those simulations for only a few values of the prevailing wind, and to fit a curve to the covariance matrices obtained to deduce covariance matrices for other prevailing winds. Notice that if additional information on the wind field is available (*e.g.*, coming from sensors onboard the UAV), it can also be incorporated in the model using, *e.g.*, a Kalman filter.

Interpolation and fitting on manifolds has been an active research topic in the past few years. In particular, in Samir *et al.* [2], the search space is infinite-

---

\*Acknowledgments: the paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office, and of the Concerted Research Action (ARC) programme supported by the Federation Wallonia-Brussels (contract ARC 14/19-060).

dimensional and the objective function is minimized with a manifold-valued gradient descent; see also Su et al. [3] for an application in image processing. Absil et al. [4] proposed interpolation techniques where the search space is restricted to a finite dimensional space of  $C^1$  piecewise-Bézier functions. We also mention the very recent work of Machado et al. [5] for the specific case of the sphere.

In this paper, as in [2], we consider a smoothing objective function—the weighted sum of a data-attachment term and a roughness penalty—but, as in [4], we restrict the search space to a class of  $C^1$  piecewise-Bézier curves. The advantage with respect to [4] is that interpolation is replaced by smoothing, a more apt framework for noisy data. The advantages with respect to [2] are (i) a lower space complexity (the solution curve is represented by a few Bézier control points on the manifold) and (ii) a considerably simpler method that only requires two objects on the manifold: the Riemannian exponential and the Riemannian logarithm. The other side of the coin is the suboptimality of the proposed approach, which is due in the first place to the restricted search space, and in the second place to the proposed computational method: it ensures optimality within the restricted search space only if the manifold is flat. Whether the suboptimality causes a significant lack of quality in practical applications is a topic for further research.

The paper is organized as follows. We first recall theory about Bézier functions generalized to manifolds and make a small reminder on the manifold arising in our application. Then, we develop our generic fitting method in Section 3, and we illustrate it on the wind field estimation problem in Section 4.

## 2 Background

**Bézier curves.** On the Euclidean space  $\mathbb{R}^r$ , *Bézier curves of degree  $K \in \mathbb{N}$*  are functions parametrized by *control points*  $b_0, \dots, b_K \in \mathbb{R}^r$  of the form

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

where  $B_{jK}(t) = \binom{K}{j} t^j (1-t)^{K-j}$  are Bernstein polynomials (also called binomial functions) [6]. One well-known way to generalize Bézier curves to a Riemannian manifold  $\mathcal{M}$  is via the De Casteljau algorithm, which only requires the Riemannian exponential and logarithm; see, e.g., [7, §2].

**The manifold  $\mathcal{S}_+(r, p)$ .** Several geometries have been proposed for the manifold  $\mathcal{S}_+(r, p)$  of  $p \times p$  PSD matrices of rank  $r$  (see [8, §7] for a survey), but to our knowledge, none of them allows turning it into a complete metric space with closed-form expressions for geodesics.

We resort here to the quotient space geometry described in [8, §7.2], in which each matrix  $S \in \mathcal{S}_+(r, p)$  is factorized as  $S = YY^T$ , where  $Y$  belongs to  $\mathbb{R}_{*}^{p \times r}$ , the set of full rank  $p \times r$  matrices. For  $Q$  orthogonal of size  $r$ , all matrices  $YQ$  are then equivalent because  $(YQ)(YQ)^T = YY^T = S$ . This geometry results in cheap closed-form expressions for end-points geodesics. The geodesic

$\gamma : [0, 1] \rightarrow \mathcal{S}_+(r, p) : t \mapsto \gamma(t)$ , with  $\gamma(0) = S_0 = Y_0 Y_0^T$  and  $\gamma(1) = S_1 = Y_1 Y_1^T$ , is given by  $\gamma(t) = \gamma_Y(t)\gamma_Y(t)^T$ , where  $\gamma_Y(t) = (1-t)Y_0 + tY_1Q^T$ , with  $Q$  the orthogonal factor of the polar decomposition  $Y_0^T Y_1 = HQ$ . In the  $Y$ -representation, the Riemannian exponential and logarithm are thus  $\exp_Y(\eta) = Y + \eta$  and  $\log_Y(Z) = ZQ'^T - Y$ , where  $\eta$  is restricted to the horizontal space  $\mathcal{H}_Y = \{\eta \in \mathbb{R}^{p \times r} : \eta^T Y = Y^T \eta\}$  and  $Q'$  comes from the polar decomposition  $Y^T Z = H'Q'$ .

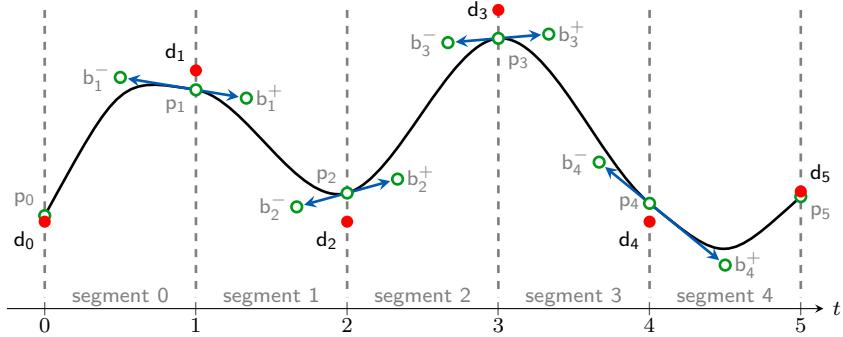
### 3 Fitting method on Riemannian manifolds

On a manifold  $\mathcal{M}$ , we aim to fit an approximating piecewise-Bézier curve  $\mathbf{B}$  to data points  $d_0, \dots, d_n$ . Its control points are chosen such that  $\mathbf{B}$  is (i) close to the data points and (ii) regular, i.e., its mean squared acceleration is small.

The piecewise-Bézier curve is composed of  $n$  Bézier functions. For notation simplicity, let  $p_0, \dots, p_n \in \mathcal{M}$  denote the endpoints of the Bézier curves and let  $(b_i^-, b_i^+) \in \mathcal{M}^2$  denote the control points on the left and right of  $p_i$ , as illustrated in Fig. 1. The first segment  $\beta^0(t) = \beta_2(t; p_0, b_1^-, p_1)$  and the last one  $\beta^{n-1}(t) = \beta_2(t; p_{n-1}, b_{n-1}^+, p_n)$  of  $\mathbf{B}$  are quadratic Bézier functions while the other segments  $\beta^i(t) = \beta_3(t; p_i, b_i^-, b_{i+1}^-, p_{i+1})$  are cubic. That being said,  $\mathbf{B}$  reads

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

Hence, continuity of  $\mathbf{B}$  is trivial. Letting  $\text{av}[(x, y), (1-\alpha, \alpha)] = \exp_x(\alpha \log_x(y))$  be a convex combination of  $x, y \in \mathcal{M}$  with weight  $\alpha \in \mathbb{R}$ , continuous differentiability is reached 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  $p_{n-1} = \text{av}[(b_{n-1}^-, b_{n-1}^+), (\frac{3}{5}, \frac{2}{5})]$ , as stated in [7, 9].



**Fig. 1:** Schematic representation of the piecewise-Bézier curve: red filled points correspond to data points, circled green ones to control points. The first and last Bézier segments are quadratic while other segments are cubic.

Ideally, we would like to compute  $\mathbf{B}$  that minimizes

$$\min_{p_i, b_i^+, b_i^-} f(\mathbf{B}) + \lambda \sum_{i=0}^n d^2(p_i, d_i), \quad (2)$$

where  $f(\mathbf{B})$  is the mean squared acceleration of  $\mathbf{B}$ . The parameter  $\lambda > 0$  adjusts the balance between data fidelity and the “smoothness” of  $\mathbf{B}$ . Note that, when  $\lambda \rightarrow \infty$ , (2) corresponds to the interpolation problem in [9]. However, instead of addressing directly this difficult optimization problem where the optimization variables are control points on  $\mathcal{M}$ , we take a suboptimal route that consists in finding optimality conditions when  $\mathcal{M} = \mathbb{R}^r$  and then generalizing those conditions to an arbitrary Riemannian manifold. As mentioned in the introduction, we will investigate in further work whether a more demanding “optimal route” would be worth the effort in practical applications.

In the easy case where  $\mathcal{M} = \mathbb{R}^r$ ,  $d(\cdot, \cdot)$  is the classical Euclidean distance and

$$f(\mathbf{B}) = \sum_{i=0}^{n-1} \int_0^1 \|\ddot{\beta}^i(t)\|^2 dt$$

Under continuity and differentiability constraints, the cost minimized in (2) is a quadratic function in the  $2n$  variables  $p_0$ ,  $(b_i^-, b_i^+)^{n-1}_{i=1}$  and  $p_n$ . Therefore, the 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,  $\mathbf{x} = [x_0, x_1, \dots, x_{2n-1}]^T := [p_0, b_1^-, b_1^+, \dots, b_{n-1}^+, p_n]^T \in \mathbb{R}^{2n \times r}$  contains the  $2n$  optimization variables and  $\mathbf{d} := [d_0, \dots, d_n]^T \in \mathbb{R}^{n+1 \times r}$  contains the data points. This problem is equivalent to  $\mathbf{x} = \lambda(A_0 + \lambda A_1)^{-1}C\mathbf{d} = Q(\lambda)\mathbf{d}$ , or to

$$x_j = \sum_{l=0}^n q_{jl}(\lambda) d_l. \quad (3)$$

Since the problem is invariant under translation, we have  $\sum_{l=0}^n q_{jl}(\lambda) = 1$  for all  $\lambda$ . In other words, the unknown control point  $x_j$  can be seen as an affine combination of the data points  $d_l$ . This also means that  $x_j - d_j^* = \sum_{l=0}^n q_{jl}(\lambda)(d_l - d_j^*)$ , by translation with respect to a reference point  $d_j^*$ . As the Euclidean difference can be seen as a logarithm map on a general manifold  $\mathcal{M}$ , a simple and natural way to generalize (3) to  $\mathcal{M}$  is to (i) use the Riemannian logarithm to map the data to the tangent space at the reference point  $d_j^*$ , (ii) compute the weighed means in the tangent space, and finally (iii) use the Riemannian exponential to map the obtained points back to the manifold  $\mathcal{M}$ . This yields

$$x_j = \exp_{d_j^*} \left( \sum_{l=0}^n q_{jl}(\lambda) \log_{d_j^*}(d_l) \right). \quad (4)$$

Observe that we recover (3) when  $\mathcal{M} = \mathbb{R}^r$ : indeed, we obtain  $x_j = d_j^* + (\sum_{l=0}^n q_{jl}(\lambda)(d_l - d_j^*))$ , where  $d_j^*$  cancels out. This cancellation does not occur in general on nonlinear manifolds; by default, we choose  $d_j^* := d_i$  for  $x_j$  being a control point  $b_i^-, b_i^+$  or  $p_i$ .

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

## 4 Numerical results

Now, we apply the proposed algorithm to the wind field estimation problem (see Figure 2). We work on a set of  $n = 33$  covariance matrices  $C(\theta_i)$  of size  $3024 \times 3024$  obtained from unsteady CFD simulations and corresponding to 33 prevailing wind orientations  $\theta_i = k\pi/64$ ,  $k \in \{0, 1, \dots, 32\}$ . For now, the magnitude of the wind field remains fixed, and further work will aim at extending our approach to develop surface fitting tools, so that both the prevailing wind magnitude and orientation are allowed to vary simultaneously. Using a singular value decomposition, we reduce the rank of  $C(\theta_i)$  to  $r = 20$  by factorizing it as  $C(\theta_i) \simeq Y(\theta_i)Y(\theta_i)^T \in \mathcal{S}_+(20, 3024)$ ; hence  $Y(\theta_i) \in \mathbb{R}_{*}^{3024 \times 20}$ . Our algorithm is implemented to directly work on the  $Y(\theta_i)$ , which has also the advantage of reducing the computational cost associated with basic matrix operations (notice that in this application, it is not even necessary to build the matrix  $C(\theta_i)$ , and one can instead obtain directly  $Y(\theta_i)$  from the simulations).

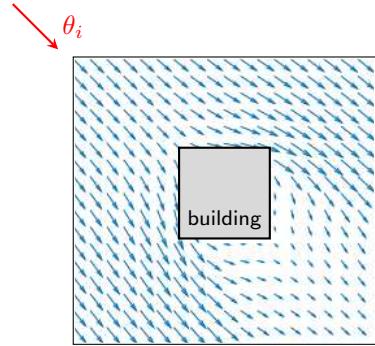
We construct the piecewise-Bézier curve  $\mathbf{B}(\theta) \in \mathcal{S}_+(20, 3024)$  based on data from a training set  $S_T = \{C(\theta_i)\}_{i \in I_T}$ , where  $I_T = \{1, 3, 5, \dots, 33\}$  and use the remaining data as a validation set  $S_V = \{C(\theta_i)\}_{i \in I_V}$ , with  $I_V = \{2, 4, \dots, 32\}$ . We measure the fitting error of  $\mathbf{B}(\theta)$  compared to data from  $S_\Omega$  ( $\Omega \in \{T, V\}$ ) as a relative mean squared error (MSE) in dB:

$$\text{MSE}(\mathbf{B}(\theta)) = 10 \log \left( \frac{\sum_{i \in I_\Omega} \|C(\theta_i) - \mathbf{B}(\theta_i)\|_F^2}{\sum_{i \in I_\Omega} \|C(\theta_i)\|_F^2} \right). \quad (5)$$

We represent its evolution with respect to the parameter  $\lambda$  in Figure 3 (left). Not surprisingly, the MSE computed on the training set decreases when  $\lambda$  grows, as problem (2) is closer and closer to interpolation. Correspondingly, the MSE computed on the validation set at the limit  $\lambda \rightarrow \infty$  measures the model error, *i.e.*, the inability of the piecewise-Bézier interpolation technique to recover the hidden data.

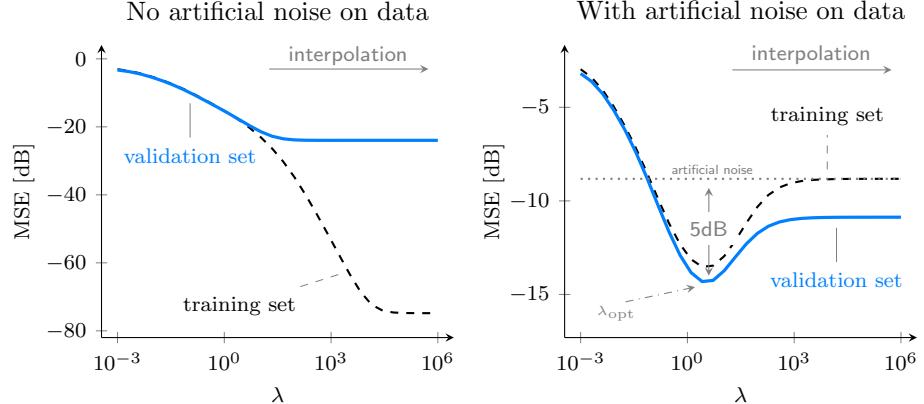
The main advantage of our method is its robustness to corrupted data. To illustrate this, we artificially added some noise to the data. Consider a new matrix  $\tilde{C}(\theta_i) = C(\theta_i) + 0.05N(\theta_i)$  with  $N(\theta_i)_{lm} \stackrel{iid}{\sim} \mathcal{N}(0, 1)$  for  $l, m = 1, \dots, 3024$ . The corrupted matrices  $\tilde{C}(\theta_i)$  are then factorized into  $\tilde{C}(\theta_i) \simeq \tilde{Y}(\theta_i)\tilde{Y}(\theta_i)^T$  similarly as above. This artificial noise results in an  $\text{MSE}(\tilde{C}(\theta))$  of about  $-9$  dB compared the (not corrupted) data points.

We compute the curve  $\tilde{\mathbf{B}}(\theta)$  based on the corrupted data from the set  $\tilde{S}_T := \{\tilde{C}(\theta_i)\}_{i \in I_T}$  and measure the MSE of  $\tilde{\mathbf{B}}(\theta)$  compared to the original data from



**Fig. 2:** Given a prevailing wind  $\theta_i$ , local wind orientation might change among the domain, specially when an object (here a building) perturbs it.

$S_\Omega$  (Figure 3, right). We observe an optimal balance  $\lambda_{\text{opt}}$  between data fitting and curve smoothing with about 5 dB of MSE reduction compared to the noise level.



**Fig. 3:** Mean squared error (MSE) obtained on the training and validation sets, without (left) or with (right) additional noise on the data. When no noise is present on the data (left), the error on the training set decreases with  $\lambda$  as the model tends to interpolation. When noise is added to data (right), our method shows denoising capacities with up to 5 dB of MSE reduction compared to the noise level.

## References

- [1] L. Pyta and D. Abel. Interpolatory galerkin models for the navier-stokes-equations. *IFAC-PapersOnLine*, 49(8):204 – 209, 2016. 2nd IFAC Workshop on Control of Systems Governed by Partial Differential Equations CPDE, 2016.
- [2] C. Samir, P.-A. Absil, A. Srivastava, and E. Klassen. A gradient-descent method for curve fitting on Riemannian manifolds. *Foundations of Computational Mathematics*, 12:49 – 73, 2012.
- [3] J. Su, I.L. Dryden, E. Klassen, H. Le, and A. Srivastava. Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds. *Image and Vision Computing*, 30(67):428 – 442, 2012.
- [4] P.-A. Absil, P.-Y. Gousenbourger, P. Striewski, and B. Wirth. Differentiable piecewise-Bézier surfaces on Riemannian manifolds. *SIAM Journal on Imaging Sciences*, to appear, 2016.
- [5] L. Machado and M. Teresa T. Monteiro. A numerical optimization approach to generate smoothing spherical splines. *Journal of Geometry and Physics*, 111:71 – 81, 2016.
- [6] G. Farin. *Curves and Surfaces for CAGD*. Academic Press, fifth edition, 2002.
- [7] T. Popiel and L. Noakes. Bézier curves and  $C^2$  interpolation in Riemannian manifolds. *Journal of Approximation Theory*, 148(2):111–127, 2007.
- [8] B. Vandereycken, P.-A. Absil, and S. Vandewalle. A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. *IMA Journal of Numerical Analysis*, 33(2):481–514, 2013.
- [9] A. Arnould, P.-Y. Gousenbourger, C. Samir, P.-A. Absil, and M. Canis. Fitting Smooth Paths on Riemannian Manifolds : Endometrial Surface Reconstruction and Preoperative MRI-Based Navigation. In F.Nielsen and F.Barbaresco, editors, *GSI2015*, 491–498. Springer International Publishing, 2015.