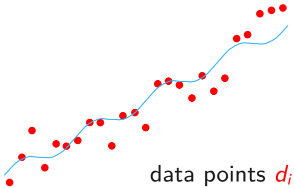
Data fitting on manifolds
applications, challenges and solutions

Pierre-Yves Gousenbourger
pierre-yves.gousenbourger@uclouvain.be

ISPGroup – Wednesday, December 11, 2019

What is the problem?

Given (t_i, d_i), find a $\mathcal{C}^1$ curve $\mathbf{B}(t)$, s.t.



data points d_i

Bézier spline!

$$\operatorname*{argmin}_{\mathbf{B}\in\Gamma} E_\lambda(\mathbf{B}) := \int_{t_0}^{t_n} \left\| \frac{\mathrm{D}^2\mathbf{B}(t)}{\mathrm{d}t^2} \right\|_{\mathbf{B}(t)}^2 \mathrm{d}t + \lambda \sum_{i=0}^{n} \mathrm{d}^2(\mathbf{B}(t_i), d_i),$$

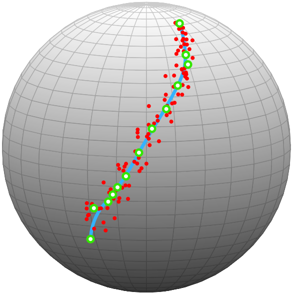
regularizer

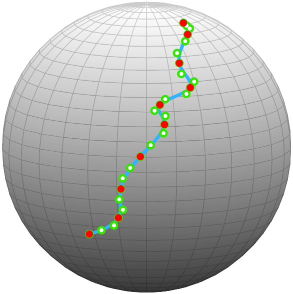
data attachment

Data points $d_i \in \mathbb{R}^2$

curve $\mathbf{B} : [0, n] \to \mathbb{R}^2$

Why is this important? – Sphere



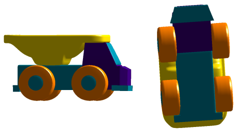


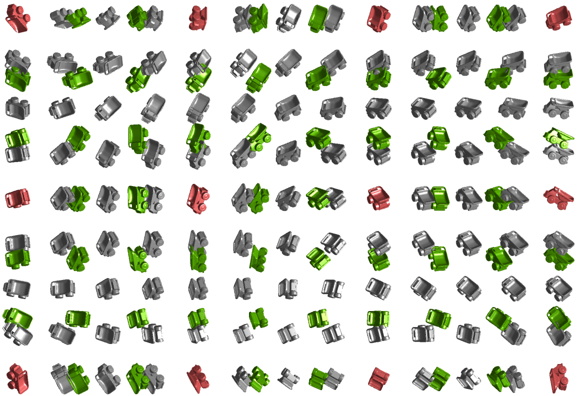
storm trajectories
birds migrations
distress planes roadmaps extrapolation

Data points $d_i \in \mathbb{S}^2$

curve $\mathbf{B} : [0, n] \to \mathbb{S}^2$

Why is this important? – Orthogonal group



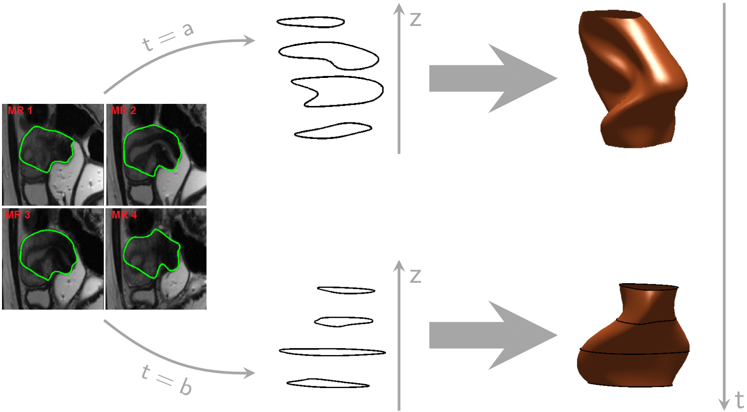


Rigid rotations of 3D objects
3D printing plannings

Computer vision, video games

Data points $d_i \in SO(3)$ curve $\mathbf{B} : [0, n] \to SO(3)$

Why is this important? - Shape space



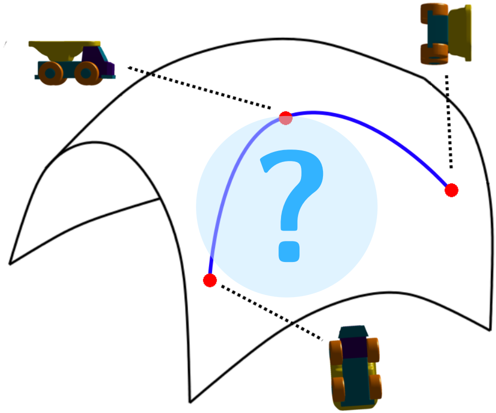
medical imaging, harmed soldiers rehab'

Data points $d_i \in \mathcal{S}$

curve $\mathbf{B} : [0, n] \to \mathcal{S}$

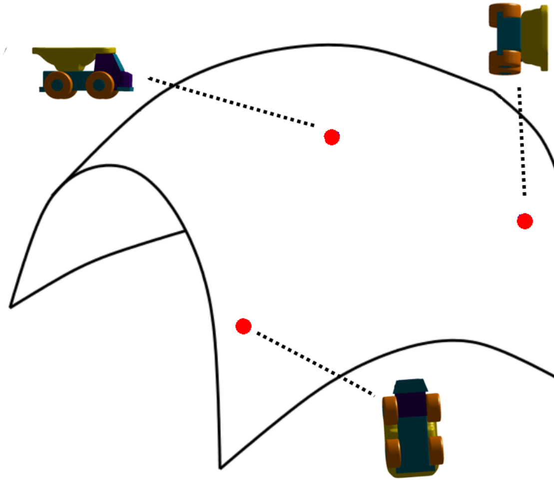
What they have in common

$\mathbb{S}^2$, $\mathrm{SO}(3)$, $\mathcal{S}_+(p, r)$, $\mathcal{S}$,... are Riemannian manifolds.

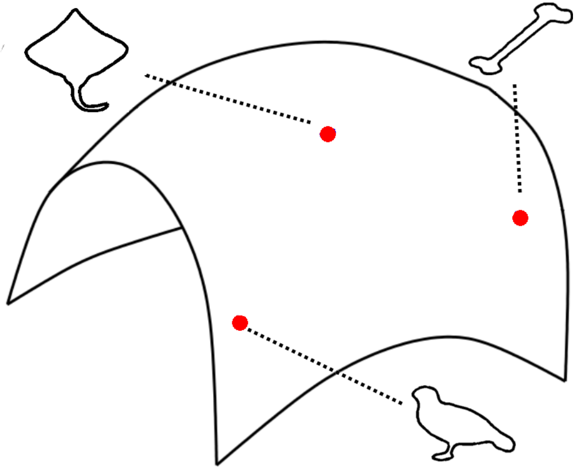


Fit • Smooth • Meaningful • Easy • Light • Fast

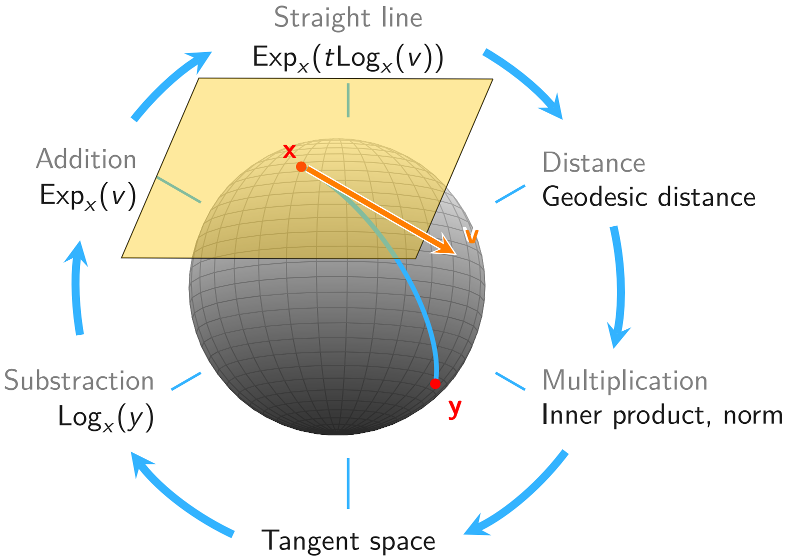
What is a manifold?



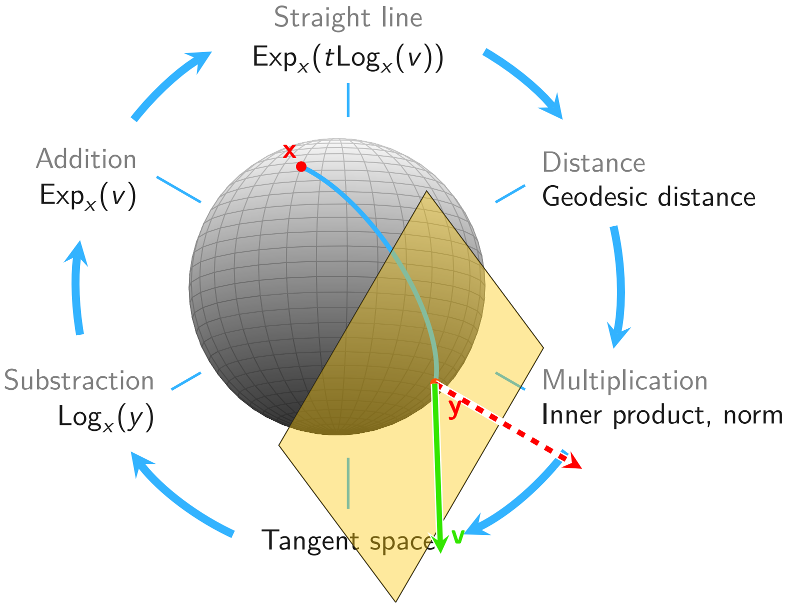
What is a manifold?



Tools of differential geometry: the sphere as an example



Tools of differential geometry: the sphere as an example

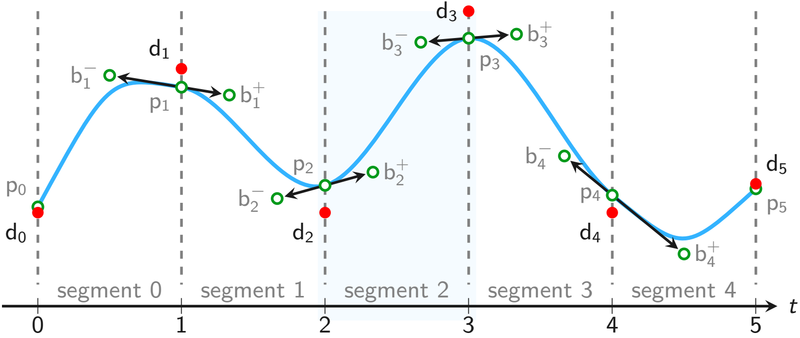


State of the art

Given data points $d_0, \ldots, d_n$ on a Riemannian manifold $\mathcal{M}$ and associated to time parameters $t_0, \ldots, t_n \in \mathbb{R}$, we seek a curve $\mathbf{B}(t)$ such that $\mathbf{B}(t_i) = d_i$.

- Geodesic regression ✗ smooth
  [Rentmeesters 2011; Fletcher 2013; Boumal 2013]
- Fitting in Sobolev space of curves ✗ fast, easy
  [Samir *et al.* 2012]
- Optimization on discretized curves ✗ light, fast
  [Boumal and Absil, 2011]
- Unrolling-unwrapping, subdivision schemes ✗ fast, easy
  [Kim 2018; Dyn 2008]

$\mathbf{B}(t)$ is a piecewise cubic Bézier curve

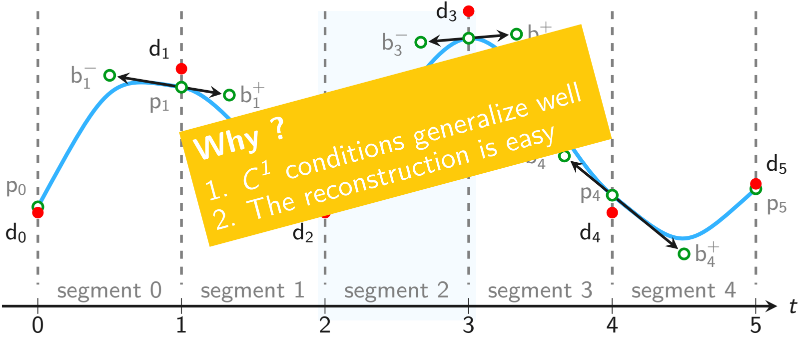


$$\mathbf{B} = \beta(t - m, \mathbf{b}^m) \text{ with } m = \lfloor t \rfloor$$

Each segment is a Bézier curve smoothly connected!

Unknowns: b_i^+, b_i^-, p_i.

$\mathbf{B}(t)$ is a piecewise cubic Bézier curve



Why ?

1. C^1 conditions generalize well
2. The reconstruction is easy

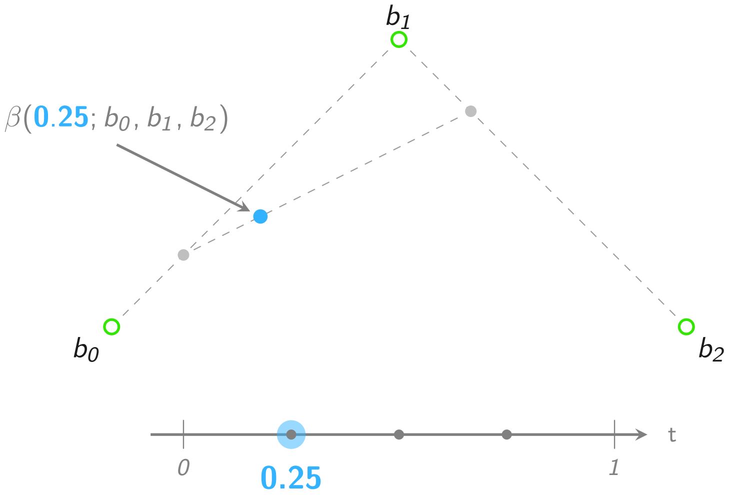
$$\mathbf{B} = \beta(t - m, \mathbf{b}^m) \text{ with } m = \lfloor t \rfloor$$

Each segment is a Bézier curve smoothly connected!

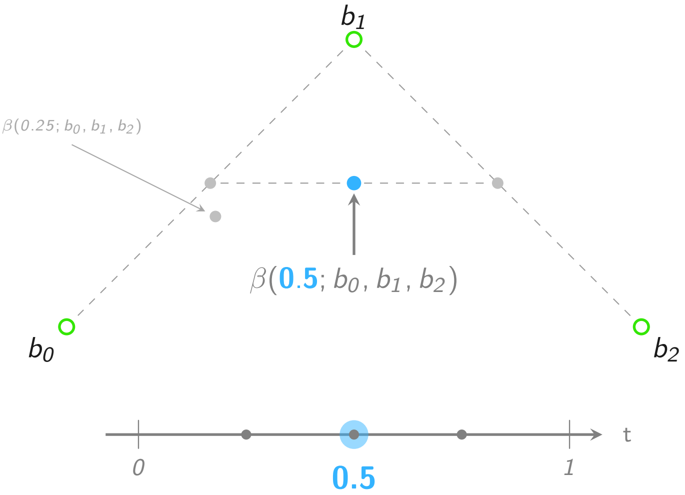
Unknowns: b_i^+, b_i^-, p_i.

C^1 conditions : $b_i^+ = g(2; b_i^-, p_i)$

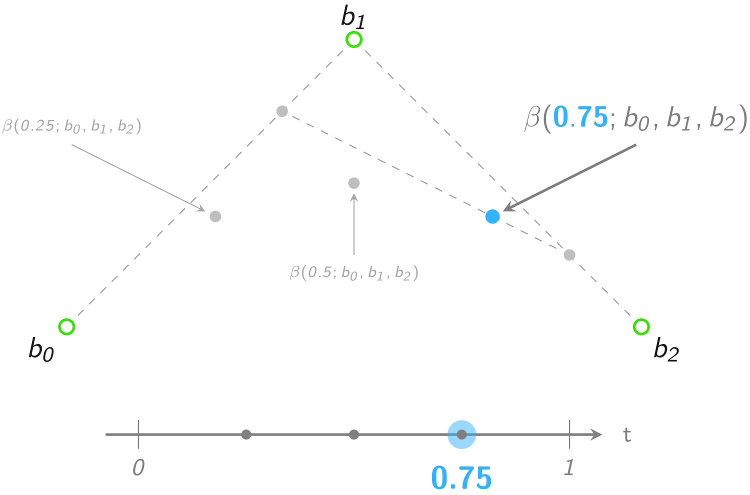
Why Bézier? – De Casteljau Algorithm generalizes well



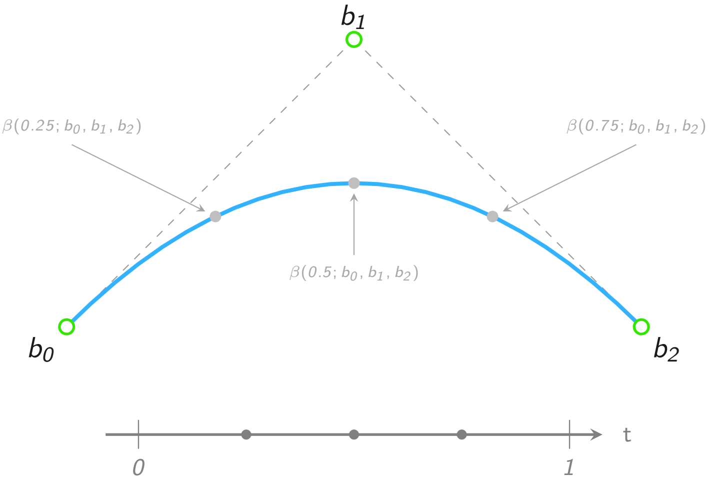
Why Bézier? – De Casteljau Algorithm generalizes well



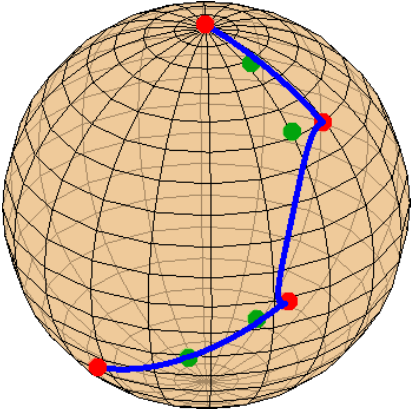
Why Bézier? – De Casteljau Algorithm generalizes well



Why Bézier? – De Casteljau Algorithm generalizes well



Why Bézier? – De Casteljau Algorithm generalizes well

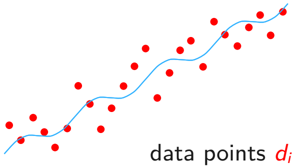


Recall: best Bezier spline to fit the data points

This is a finite dimensionnal optimization problem in b_i^-, p_i.

The goal:

- Find the minimizer $\mathbf{B}$ (on $\mathcal{M} = \mathbb{R}^n$: natural cubic spline).
- What is the gradient ?



Fitting curve

$$\operatorname*{argmin}_{\mathbf{B} \in \Gamma} E_\lambda(\mathbf{B}) := \int_{t_0}^{t_n} \left\| \frac{\mathrm{D}^2 \mathbf{B}(t)}{\mathrm{d}t^2} \right\|_{\mathbf{B}(t)}^2 \mathrm{d}t + \lambda \sum_{i=0}^{n} \mathrm{d}^2(\mathbf{B}(t_i), d_i),$$

this is just another geodesic...

???!!!

How to compute the control points?

in $\mathbb{R}^d$

Unique $\mathcal{C}^2$ smoothing polynomial spline

$$\text{s.t. } \min_{b_i^m} \int_0^M \|\mathbf{B}''(t)\| \mathrm{d}t$$

(Long story short)

$$b_i^m = \sum_{j=0}^{n} q_{i,j} d_j$$

Generalization to $\mathcal{M}$

in $\mathcal{M}$?

1. Invariance to translation to a point d_{ref}.
2. Translation to d_{ref} is a Riemannian Log on $\mathbb{R}^r$.
3. Exponentiall map to go back to $\mathcal{M}$.
4. Compute p_i with the manifold-valued $\mathcal{C}^1$ condition.

$$b_i^m - d_{\text{ref}} = \sum_{j=0}^{n} q_{i,j}(d_j - d_{\text{ref}})$$

Generalization to $\mathcal{M}$

in $\mathcal{M}$?

1. Invariance to translation to a point d_{ref}.
2. Translation to d_{ref} is a Riemannian Log on $\mathbb{R}^r$.
3. Exponentiall map to go back to $\mathcal{M}$.
4. Compute p_i with the manifold-valued $\mathcal{C}^1$ condition.

$$\log_{d_{\text{ref}}}(b_i^m) = \sum_{j=0}^{n} q_{i,j} \log_{d_{\text{ref}}}(d_j)$$

Generalization to $\mathcal{M}$

in $\mathcal{M}$?

1. Invariance to translation to a point d_{ref}.
2. Translation to d_{ref} is a Riemannian Log on $\mathbb{R}^r$.
3. Exponential map to go back to $\mathcal{M}$.
4. Compute p_i with the manifold-valued $\mathcal{C}^1$ condition.

$$b_i^m = \exp_{d_{\text{ref}}}\left(\tilde{b}_i^m\right)$$

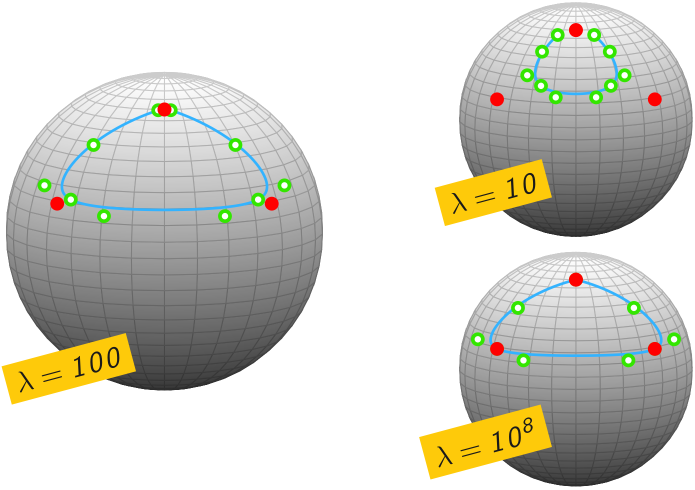
Generalization to $\mathcal{M}$

in $\mathcal{M}$?

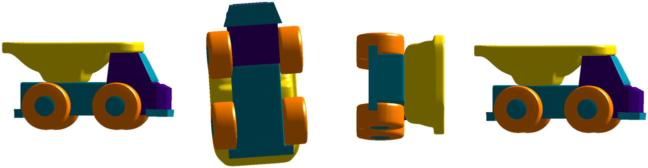
1. Invariance to translation to a point d_{ref}.
2. Translation to d_{ref} is a Riemannian Log on $\mathbb{R}^r$.
3. Exponential map to go back to $\mathcal{M}$.
4. Compute p_i with the manifold-valued $\mathcal{C}^1$ condition.

$$p_i = g(0.5; b_i^-, b_i^+)$$

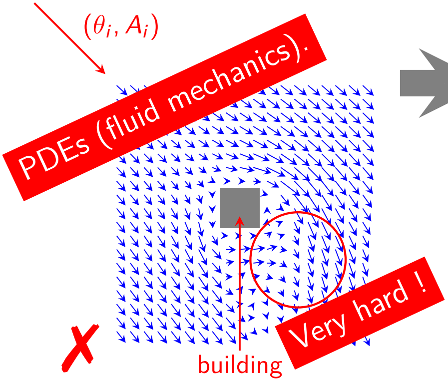
Results show that it works... ($\mathbb{S}^2$)



Results show that it works... (SO(3))

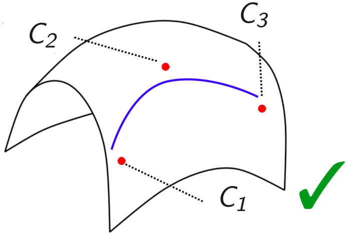


... but it actually fails sometimes ($\mathcal{S}_+(p, r)$)



$W_i = W(\theta_i, A_i) \sim \mathcal{N}(\mu_i, C_i)$

$C_i \in \mathcal{S}_+(p, r)$



Wind field estimation for UAV

Data points $d_i \in \mathcal{S}_+(p, r)$

curve **B** : $[0, n] \to \mathcal{S}_+(p, r)$

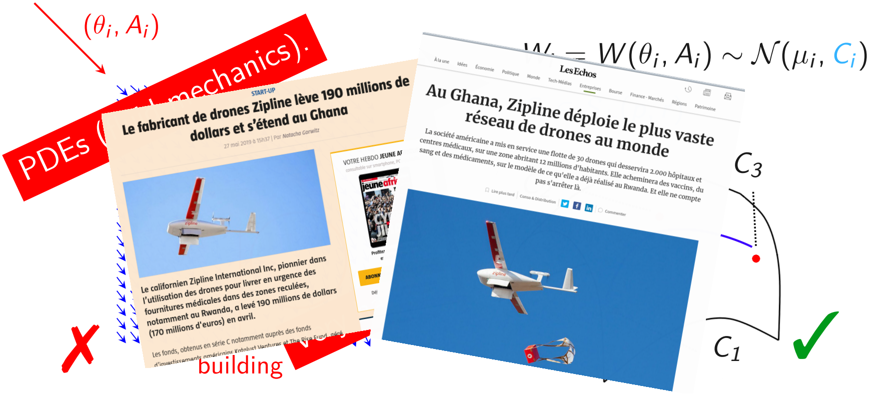
... but it actually fails sometimes ($\mathcal{S}_+(p, r)$)

Wind field estimation for UAV

Data points $d_i \in \mathcal{S}_+(p, r)$

curve **B** : $[0, n] \to \mathcal{S}_+(p, r)$

... but it actually fails sometimes ($\mathcal{S}_+(p, r)$)

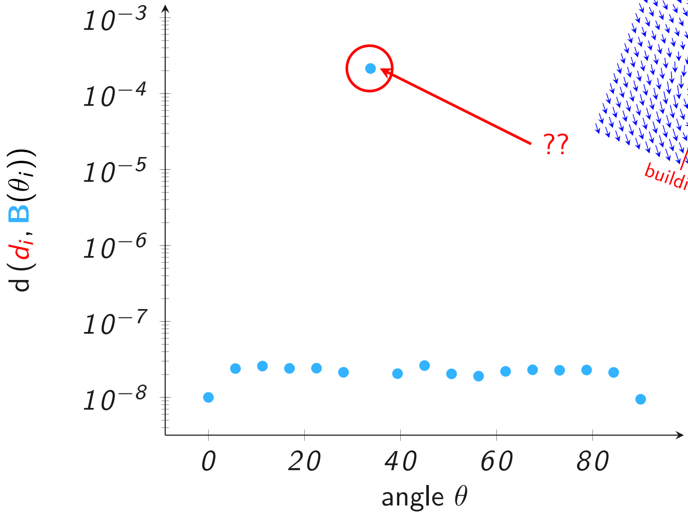


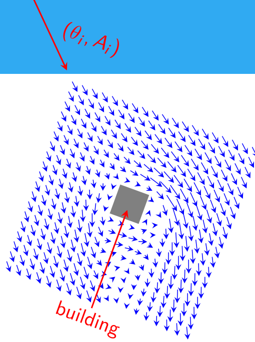
Wind field estimation for UAV

Data points $d_i \in \mathcal{S}_+(p, r)$

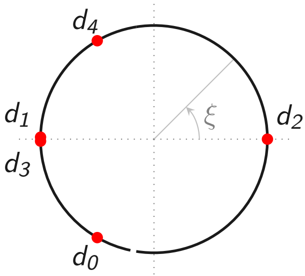
curve $\mathbf{B} : [0, n] \to \mathcal{S}_+(p, r)$

... but it actually fails sometimes ($\mathcal{S}_+(p, r)$)





... but it actually fails sometimes ($\mathbb{S}^1$)

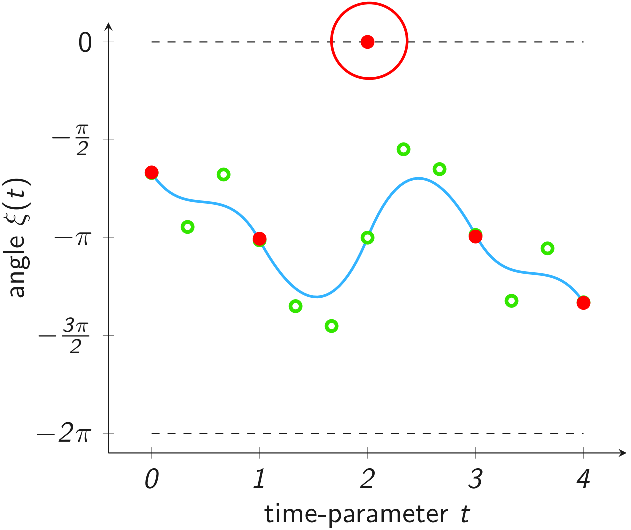


$(x, y) = (\cos\xi, \sin\xi)$

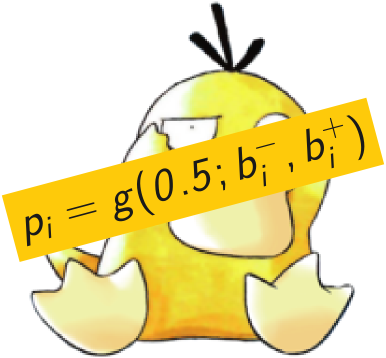
Data points $d_i \in \mathbb{S}^1$

curve **B** : $[0, 4] \to \mathbb{S}^1$

... but it actually fails sometimes ($\mathbb{S}^1$)

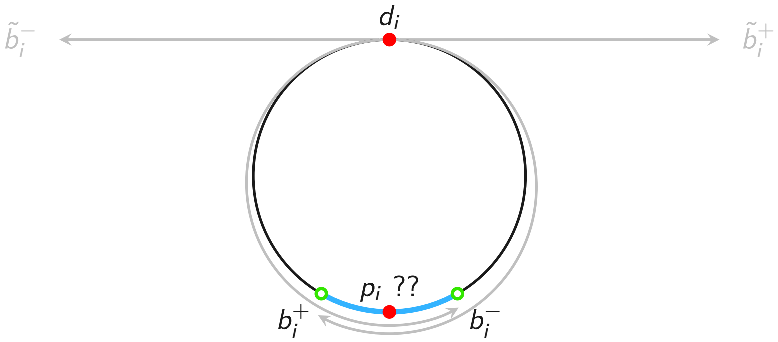


So what's wrong?

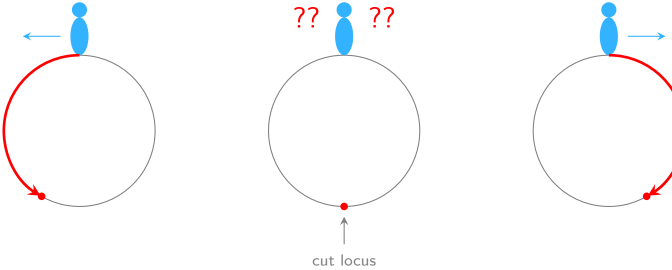


$$p_i = g(0.5; b_i^-, b_i^+)$$

The failure revealed in the $\mathcal{C}^1$ condition! ($\mathbb{S}^1$)



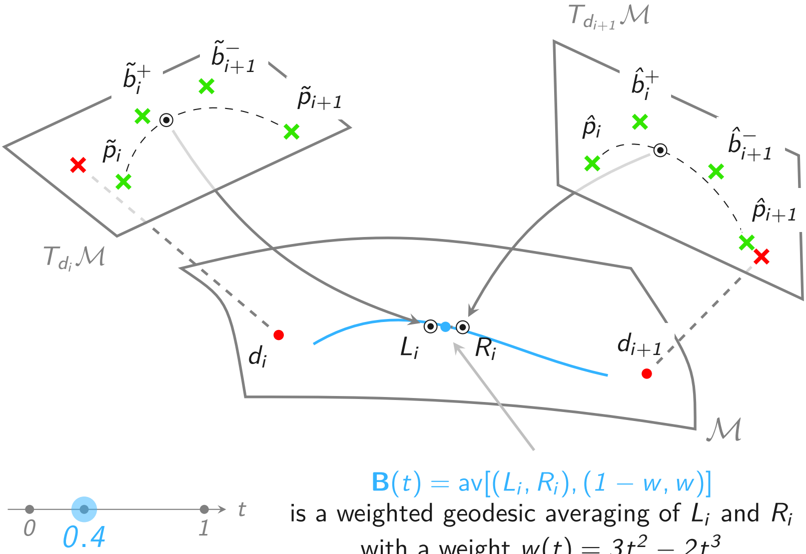
The curse of the curvature: the cut locus



The injectivity radius r_{inj} of $\mathcal{M}$ is the smallest max-distance between two points such that the cut locus is never crossed.

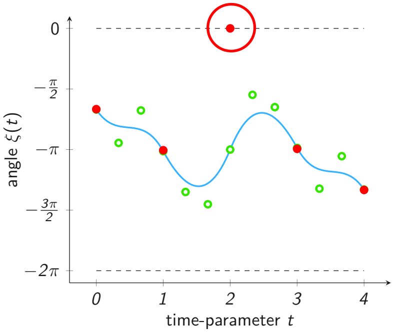
Ex: $r_{inj}(\mathbb{S}^1) = \pi$.

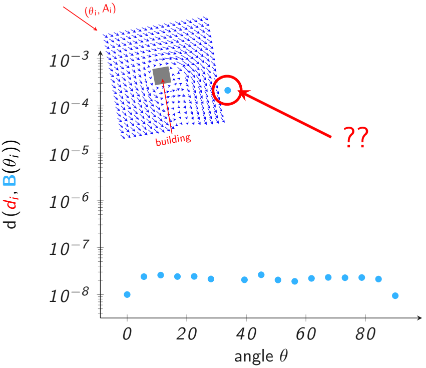
The Cubic Blended Splines Algorithm



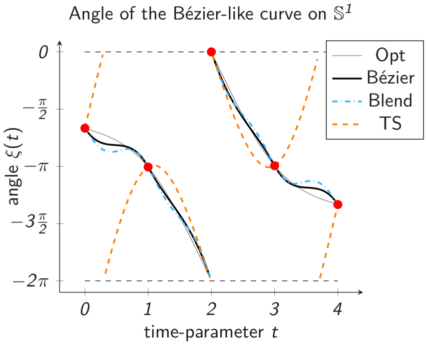
$\mathbf{B}(t) = \mathrm{av}[(L_i, R_i), (1-w, w)]$
is a weighted geodesic averaging of L_i and R_i
with a weight $w(t) = 3t^2 - 2t^3$

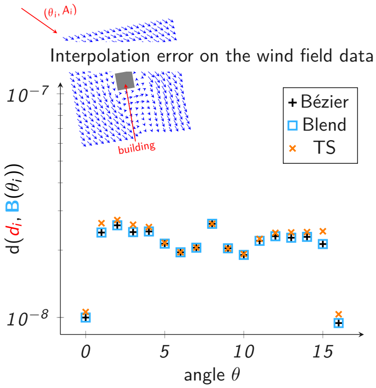
Results: before...





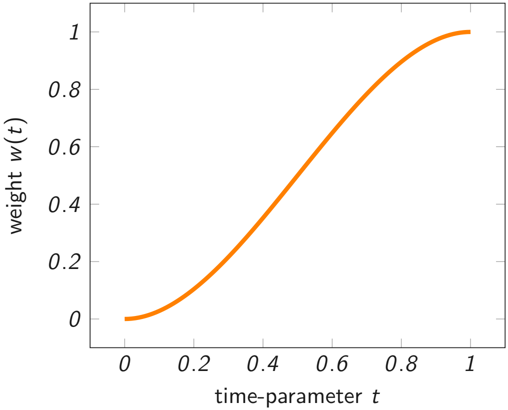
Results: … after





Properties

1. $\mathbf{B}(t_i) = d_i$ when $\lambda \to \infty$;
2. $\mathbf{B}(t)$ is $\mathcal{C}^1$; $(\beta_i(t) = \mathrm{av}[(L_i, R_i), (1 - w(t), w(t))])$
3. $\mathbf{B}(t)$ is the natural smoothing spline when $\mathcal{M} = \mathbb{R}^r$

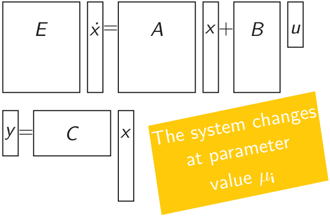


Properties

1. $\mathbf{B}(t_i) = d_i$ when $\lambda \to \infty$;
2. $\mathbf{B}(t)$ is $\mathcal{C}^1$;
3. $\mathbf{B}(t)$ is the natural smoothing spline when $\mathcal{M} = \mathbb{R}^r$
4. Only Exp and Log involved. No optimization.
5. Only $\mathcal{O}(n)$ tangent vectors to store;
6. Only $\mathcal{O}(1)$ operations to evaluate $\mathbf{B}(t)$ at $t \in [0, 1]$.

Fit • Smooth • Meaningful • Easy • Light • Fast

Comparison with other fitting techniques (PMOR)



1. Find the low rank solutions

$P_i = X_i X_i^\top, \quad Q_i = Y_i Y_i^\top$

of the Lyapunov equations:

$$\begin{cases} E P_i A^\top + A P_i E^\top = -BB^\top \\ E^\top Q_i A + A^\top Q_i E = -C C^\top \end{cases}$$

$X_i \in \mathbb{R}^{n \times k_{X_i}}$, $Y_i \in \mathbb{R}^{n \times k_{Y_i}}$ — truncate! →

$P_i \in \mathcal{S}_+(p, n)$, $\quad p = \min_i(k_{X_i})$

$Q_i \in \mathcal{S}_+(q, n)$, $\quad q = \min_i(k_{Y_i})$

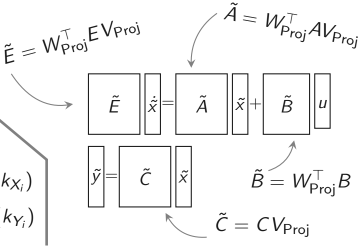
2. Find projectors V_{Proj} and W_{Proj} with an SVD

2.1. SVD step: $Y_i^\top E X_i = U_i \Sigma_i V_i^\top$

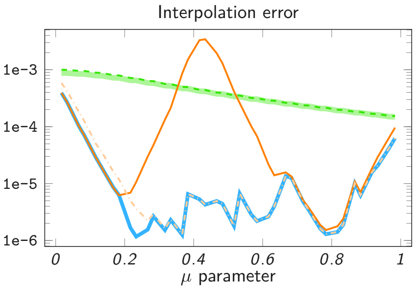
2.2. Σ_i truncated as $\tilde{\Sigma}_i$ to r largest values.
V_i and U_i truncated to r first columns $\tilde{V}_i$ and $\tilde{U}_i$.

2.3. $V_{\text{Proj}} = X_i \tilde{V}_i \tilde{\Sigma}_i^{-\frac{1}{2}}$ and $W_{\text{Proj}} = Y_i \tilde{U}_i \tilde{\Sigma}_i^{-\frac{1}{2}}$

3. **The end-goal:** reduce the model



Comparison with other fitting techniques (PMOR)



$$E_{\mathrm{rel}}(\mu) = \frac{\|P(\mu) - P^{\mathrm{GT}}(\mu)\|_{\mathrm{F}}}{\|P^{\mathrm{GT}}(\mu))\|_{\mathrm{F}}}$$

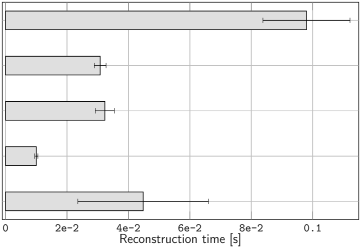
Blending method (cubic)

TS method at Q_1 (cubic)

TS method at Q_{mid} (cubic)

Piecewise-geodesic interpolation

Piecewise linear interpolation in $\mathbb{R}^{n\times n}$



Conclusions and future work

Take-home message:

- Smooth fitting method: fast, efficient, portable;
- The cut locus is still a curse;
- Tangent-space based methods are close to optimal for "local" data points;

Future work:

- Generalization to **2D**, **3D**, is open (ongoing with B.Wirth, Universität Münster);
- Theoretical bound on the suboptimality: open question.

Data fitting on manifolds
applications, challenges and solutions

Pierre-Yves Gousenbourger

pierre-yves.gousenbourger@uclouvain.be

📝 **G., Massart and Absil.** *Data fitting on manifolds with composite Bézier-like curves and blended cubic splines*. Journal of Mathematical Imaging and Vision, 61(5), pp. 645–671, 2018.

Code available on perso.uclouvain.be/pygousenbourger/