Optimization on matrix manifolds and applications

Marco Sutti

Postdoctoral fellow at NCTS

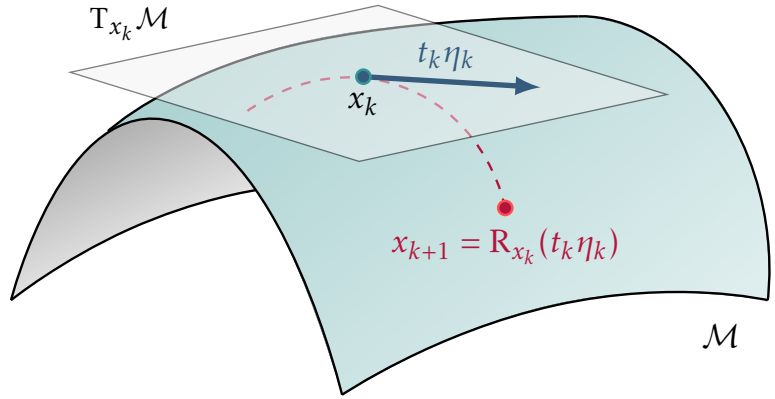
國家理論科學研究中心 數學組、博士後

NCU Mathematics Colloquium

December 12, 2024

Overview

- Numerical algorithms for optimization on **matrix manifolds**.
- Exploit **geometric structure**, take into account the constraints.
- **Framework** that encompasses almost all of my research works.



This talk:

I. **Numerical (Riemannian) optimization on matrix manifolds**, fundamental ideas and tools.

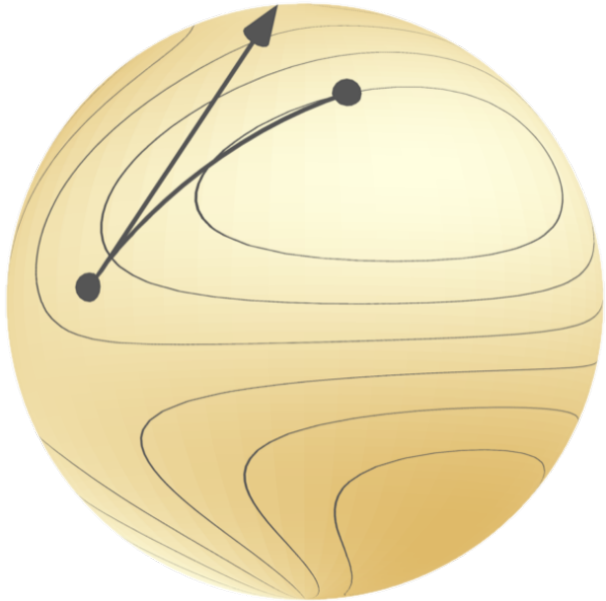
II. **Applications** on:

- the manifold of **fixed-rank matrices** (low-rank time integration of a PDE);
- the **power manifold of unit spheres** (computer graphics);
- the **Stiefel manifold** (interpolation on manifolds).

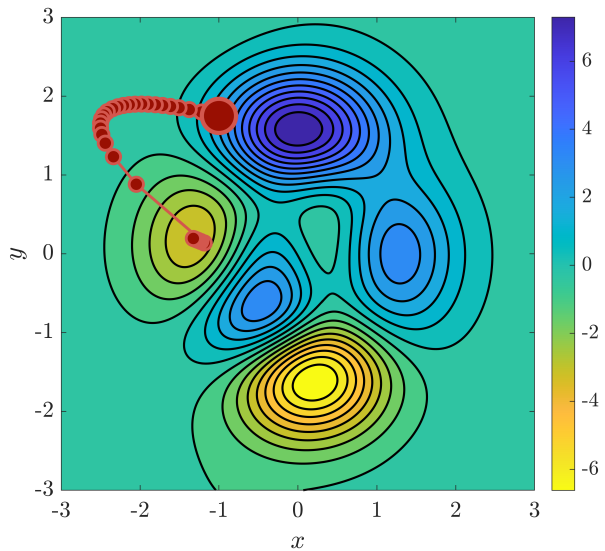
I. Optimization on matrix manifolds

Riemannian optimization/1

- The Riemannian optimization framework solves constrained optimization problems where the constraints have a geometric nature.
  - Exploit the underlying geometric structure of the problems. The optimization variables are constrained to a smooth manifold.

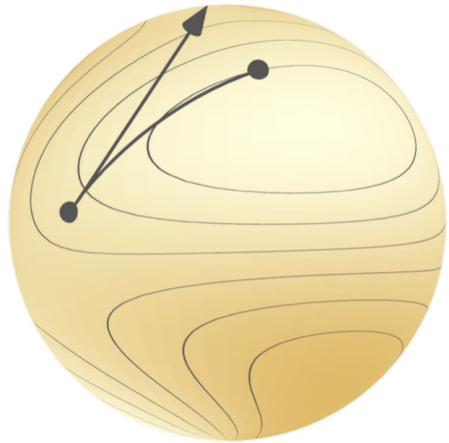


- Traditional optimization methods rely on the Euclidean vector space structure.
  - E.g., the steepest descent method for minimizing $f : \mathbb{R}^n \to \mathbb{R}$ updates $\mathbf{x}_k$ by moving in the direction $\mathbf{d}_k$ of the anti-gradient of f, by a step size α_k chosen according to a line-search rule.



Manifold optimization: [Edelman et al. 1998, Absil et al. 2008, Boumal 2023], ...
The image above has been taken from the Manopt website: https://www.manopt.org/

Riemannian optimization/2



- Formally, we can state the optimization problem as

$$\min_{x \in \mathcal{M}} f(x),$$

  where $f : \mathcal{M} \to \mathbb{R}$ is the objective function and $\mathcal{M}$ is some matrix manifold.

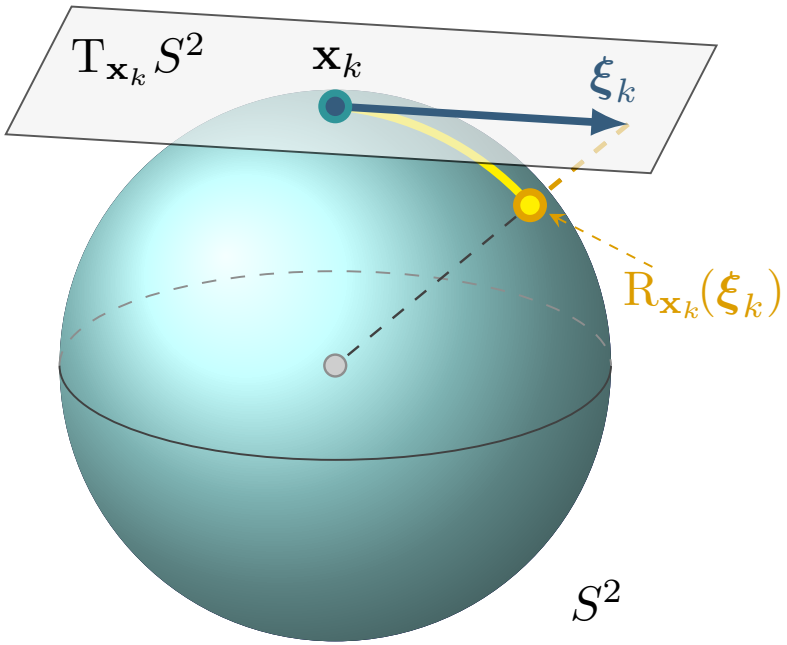
- **Matrix manifold**: any manifold that is constructed from $\mathbb{R}^{n \times p}$ by taking either **embedded submanifolds** or **quotient manifolds**.
  - Examples of embedded submanifolds: unit sphere, orthogonal Stiefel manifold, manifold of fixed-rank matrices, …
  - Examples of quotient manifolds: the Grassmann manifold, the flag manifold.
- A manifold $\mathcal{M}$ endowed with a smoothly-varying inner product (called Riemannian metric g) is called Riemannian manifold.

  $\rightsquigarrow$ A couple $(\mathcal{M}, g)$, i.e., a manifold with a Riemannian metric on it.

Manifold optimization: [Edelman et al. 1998, Absil et al. 2008, Boumal 2023], …

Riemannian optimization/3

- A line-search method in the Riemannian framework determines at $\mathbf{x}_k$ on a manifold $\mathcal{M}$ a search direction $\boldsymbol{\xi}_k$ on $\mathrm{T}_{\mathbf{x}_k}\mathcal{M} \to \mathcal{M}$.
- $\mathbf{x}_{k+1}$ is then determined by a line search along a curve $\alpha \mapsto \mathrm{R}_{\mathbf{x}_k}(\alpha\boldsymbol{\xi}_k)$ where $\mathrm{R}_{\mathbf{x}_k} : \mathrm{T}_{\mathbf{x}_k}\mathcal{M} \to \mathcal{M}$ is the retraction mapping.
- Repeat for $\mathbf{x}_{k+1}$ in the role of $\mathbf{x}_k$.



- Search directions can be the negative of the Riemannian gradient, leading to the Riemannian gradient descent method (RGD).
  - Other choices of search directions $\rightsquigarrow$ other methods, e.g., **Riemannian trust-region method** or **Riemannian BFGS**.

Riemannian trust-region method: [Absil/Baker/Gallivan 2007], Riemannian BFGS: [Ring/Wirth 2012]

The manifold of fixed-rank matrices $\mathcal{M}_r$

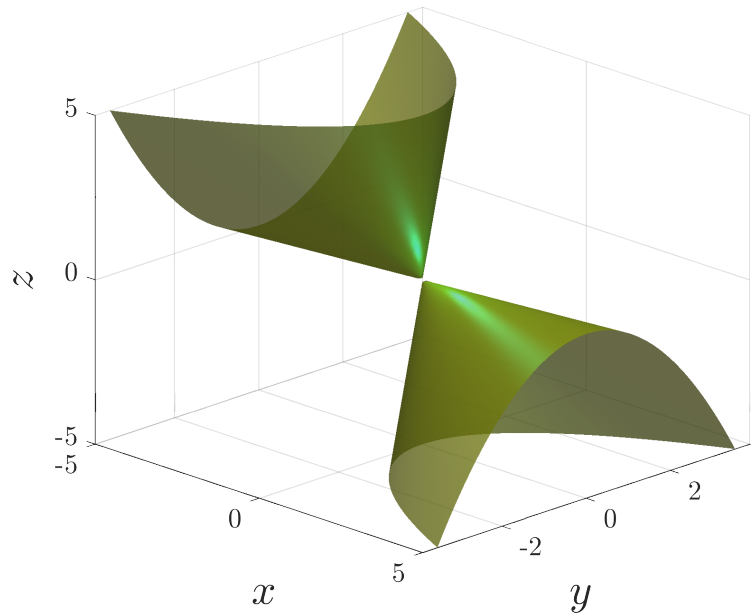
- Later, we will define an optimization problem over

$$\mathcal{M}_r = \{X \in \mathbb{R}^{m\times n} : \ \mathrm{rank}(X) = r\}.$$

$\rightsquigarrow \mathcal{M}_r$ has a smooth structure …

2×2 example:

$$X = \begin{bmatrix} x & -2y \\ y & z \end{bmatrix}.$$



Parametrization:
$\mathrm{rank}(X) = 1 \Leftrightarrow xz = -2y^2$ and
$x, z \neq 0$.

- Using the SVD, one has the equivalent characterization

$$\mathcal{M}_r = \{U\Sigma V^\top : \ U \in \mathrm{St}(m,r), \ V \in \mathrm{St}(n,r), \\ \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}, \ \sigma_1 \geq \cdots \geq \sigma_r > 0\}.$$

Optimizing on submanifold $\mathcal{M}_r$: [Vandereycken 2013]

The unit sphere S^2 and the power manifold $\left(S^2\right)^n$

The unit sphere S^2 is a Riemannian submanifold of $\mathbb{R}^3$ defined as

$$S^2 = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x}^\top \mathbf{x} = 1\}.$$

The Riemannian metric on the unit sphere is inherited from $\mathbb{R}^3$, i.e.,

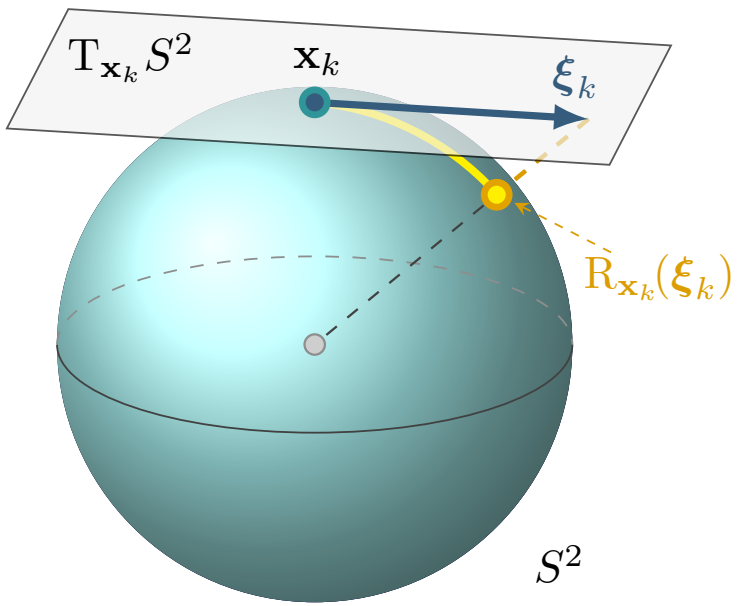
$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\mathbf{x}} = \boldsymbol{\xi}^\top \boldsymbol{\eta}, \quad \boldsymbol{\xi}, \boldsymbol{\eta} \in \mathrm{T}_{\mathbf{x}} S^2,$$

where $\mathrm{T}_{\mathbf{x}} S^2$ is the tangent space to S^2 at $\mathbf{x} \in S^2$, defined as the set of all vectors orthogonal to $\mathbf{x}$ in $\mathbb{R}^3$, i.e.,

$$\mathrm{T}_{\mathbf{x}} S^2 = \{\mathbf{z} \in \mathbb{R}^3 : \mathbf{x}^\top \mathbf{z} = 0\}.$$

The projector $\mathrm{P}_{\mathrm{T}_{\mathbf{x}} S^2} : \mathbb{R}^3 \to \mathrm{T}_{\mathbf{x}} S^2$

$$\mathrm{P}_{\mathrm{T}_{\mathbf{x}} S^2}(\mathbf{z}) = (I_3 - \mathbf{x}\mathbf{x}^\top)\mathbf{z}.$$

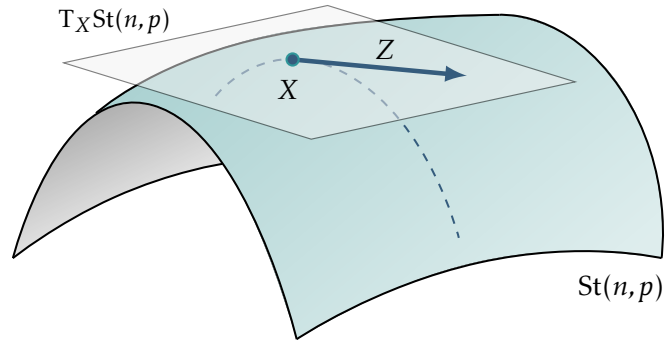


$\rightsquigarrow$ Later, we will consider the power manifold of n unit spheres $\left(S^2\right)^n = \underbrace{S^2 \times S^2 \times \cdots S^2}_{n \text{ times}}$, with the metric of S^2 extended elementwise.

The Stiefel manifold $\mathrm{St}(n,p)$

- Set of matrices with orthonormal columns:

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n\times p} : \ X^\top X = I_p\}.$$



- Tangent space to $\mathcal{M}$ at x: set of all tangent vectors to $\mathcal{M}$ at x, denoted $\mathrm{T}_x\mathcal{M}$. For $\mathrm{St}(n,p)$,

$$\mathrm{T}_X\mathrm{St}(n,p) = \{\xi \in \mathbb{R}^{n\times p} : \ X^\top \xi + \xi^\top X = 0\}.$$

- Alternative characterization of $\mathrm{T}_X\mathrm{St}(n,p)$:

$$\mathrm{T}_X\mathrm{St}(n,p) = \{X\Omega + X_\perp K : \ \Omega = -\Omega^\top, \ K \in \mathbb{R}^{(n-p)\times p}\},$$

where $\mathrm{span}(X_\perp) = \Big(\mathrm{span}(X)\Big)^\perp$.

- The projection onto the tangent space $\mathrm{T}_X\mathrm{St}(n,p)$ is

$$\mathrm{P}_X\,\xi = X\mathrm{skew}(X^\top \xi) + (I - XX^\top)\,\xi.$$

Stiefel manifold: [Stiefel, 1935]

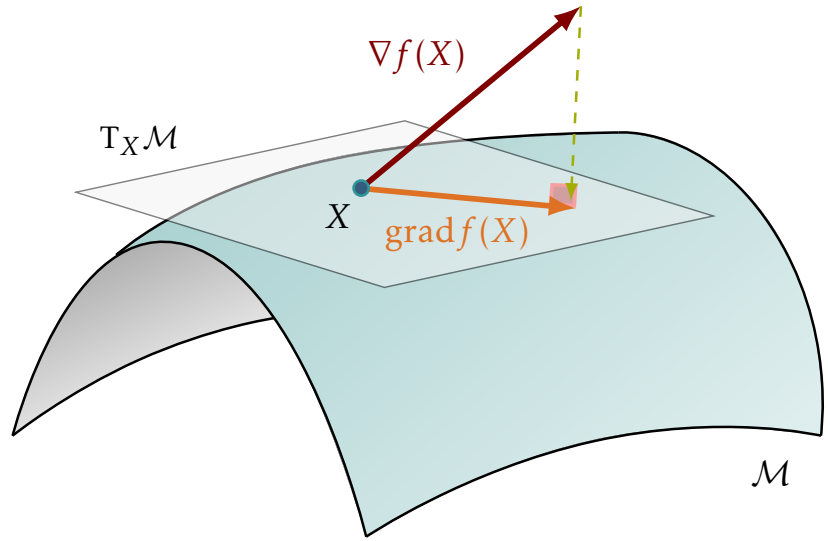
Riemannian gradient

Let $f : \mathcal{M} \to \mathbb{R}$. E.g., the objective function in an optimization problem.

$\rightsquigarrow$ For any embedded submanifold (Prop. 3.6.1 in Absil et al., 2008):

- Riemannian gradient: projection onto $\mathrm{T}_X\mathcal{M}$ of the **Euclidean gradient**

$$\operatorname{grad} f(X) = \mathrm{P}_{\mathrm{T}_X\mathcal{M}}(\nabla f(X)).$$



$\rightsquigarrow$ Recall: orthogonal projection onto the tangent space to

the unit sphere:

$$\mathrm{P}_{\mathrm{T}_{\mathbf{x}}S^2}(\mathbf{z}) = (I - \mathbf{x}\mathbf{x}^\top)\mathbf{z}.$$

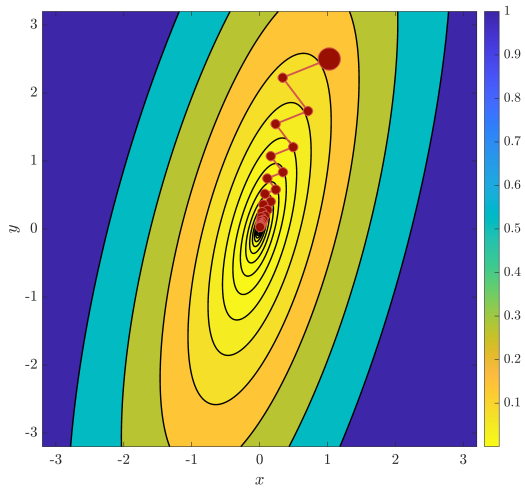
the Stiefel manifold:

$$\mathrm{P}_X\,\xi = X\operatorname{skew}(X^\top\xi) + (I - XX^\top)\xi.$$

$\rightsquigarrow$ $\nabla f(X)$ is the **Euclidean gradient** of $f(X)$.

Matrix and vector calculus: The Matrix Cookbook, www.matrixcalculus.org, …
Automatic differentiation on low-rank manifolds: [Novikov/Rakhuba/Oseledets 2022]

Steepest descent on a manifold



- Steepest descent in $\mathbb{R}^n$ is based on the update formula

$$x_{k+1} = x_k + t_k \eta_k,$$

where $t_k \in \mathbb{R}$ is the step size and $\eta_k \in \mathbb{R}^n$ is the search direction.

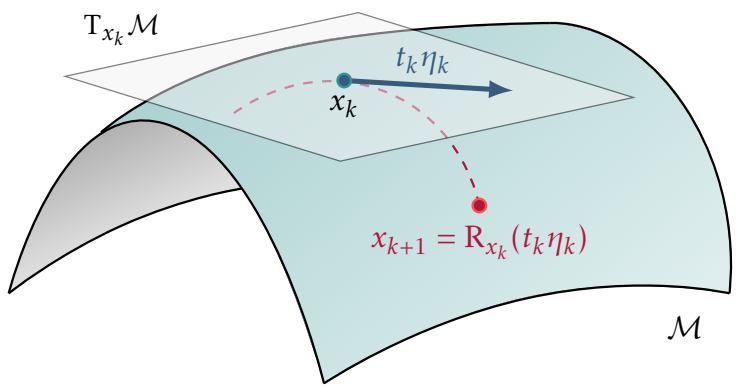
$\rightsquigarrow$ On nonlinear manifolds:

- η_k will be a tangent vector to $\mathcal{M}$ at x_k, i.e., $\eta_k \in \mathrm{T}_{x_k}\mathcal{M}$.

  Remark: If $\eta_k = -\operatorname{grad} f(x_k)$, we get the **Riemannian steepest descent**.

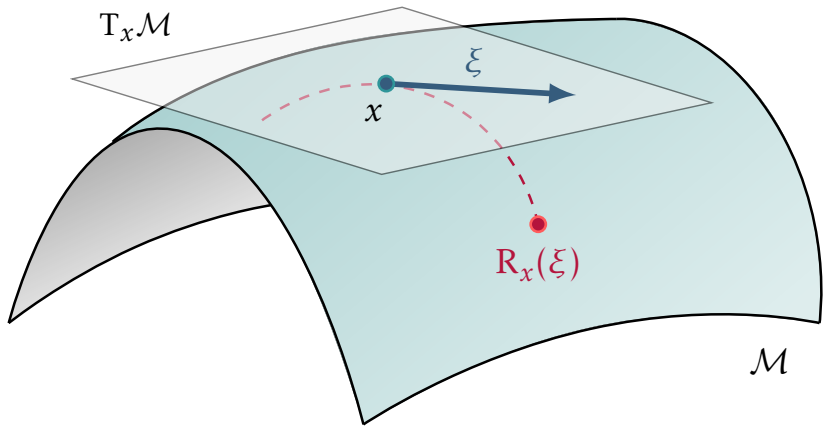
- Search along a curve in $\mathcal{M}$ whose tangent vector at $t_k = 0$ is η_k.

$\rightsquigarrow$ Retraction.



Retractions

- Move in the direction of ξ while remaining constrained to $\mathcal{M}$.
- Smooth mapping $\mathrm{R}_x : \mathrm{T}_x\mathcal{M} \to \mathcal{M}$ with a local condition that preserves gradients at x.



- The Riemannian exponential mapping is also a retraction, but it is not computationally efficient.
- Retractions: first-order approximation of the Riemannian exponential!

Constructing retractions: [Absil/Malick 2012]

Steepest descent on a manifold (reprise)

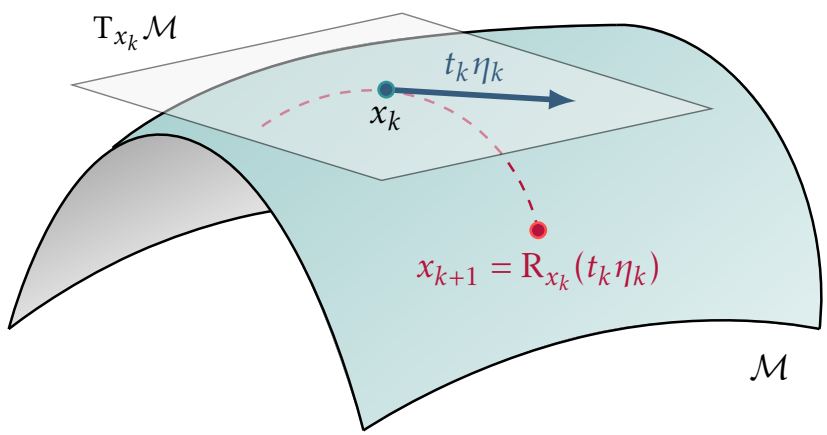
Steepest descent on manifolds is based on the update formula

$$x_{k+1} = \mathrm{R}_{x_k}(t_k \eta_k),$$

where $t_k \in \mathbb{R}$ and $\eta_k \in \mathrm{T}_{x_k}\mathcal{M}$.

Recipe for constructing the steepest descent method on a manifold:

- Choose a retraction R (previous slide).
- Select a search direction η_k (the anti-gradient $\eta_k = -\operatorname{grad} f(x_k)$).
- Select a step length t_k (with a line-search technique).



II. Research works and applications

Implicit low-rank Riemannian schemes

Paper: Implicit low-rank Riemannian schemes for the time integration of stiff partial differential equations, M. Sutti and B. Vandereycken, Vol. 101, article number 3, J. Sci. Comput., 13 August 2024.

Main contributions:

- Preconditioner for the Riemannian trust-region (RTR) method on the manifold of fixed-rank matrices.
- Applications within implicit numerical integration schemes to solve stiff, time-dependent PDEs.

This part of the talk:

I. Motivation for considering the low-rank format.

II. Riemannian trust-region method and preconditioning.

III. The Allen–Cahn equation.

Motivation for the low-rank format

- Often, like in computational fluid dynamics (CFD), we need to discretize a problem to represent the continuous solution.
- For high-dimensional problems (e.g., Schrödinger equation, Black–Scholes equation…), a "naive" discretization with n degrees of freedom in each dimension leads to n^d coefficients.
- For example, if $d = 15$, and $n = 100$ grid points in each dimension, we would need 8 000 TB of memory to store all coefficients in double precision:

$$\frac{100^{15} \times 64 \text{ bits}}{8 \times 10^{12}} = 8 \times 10^{18} = 8\,000 \text{ TB}.$$

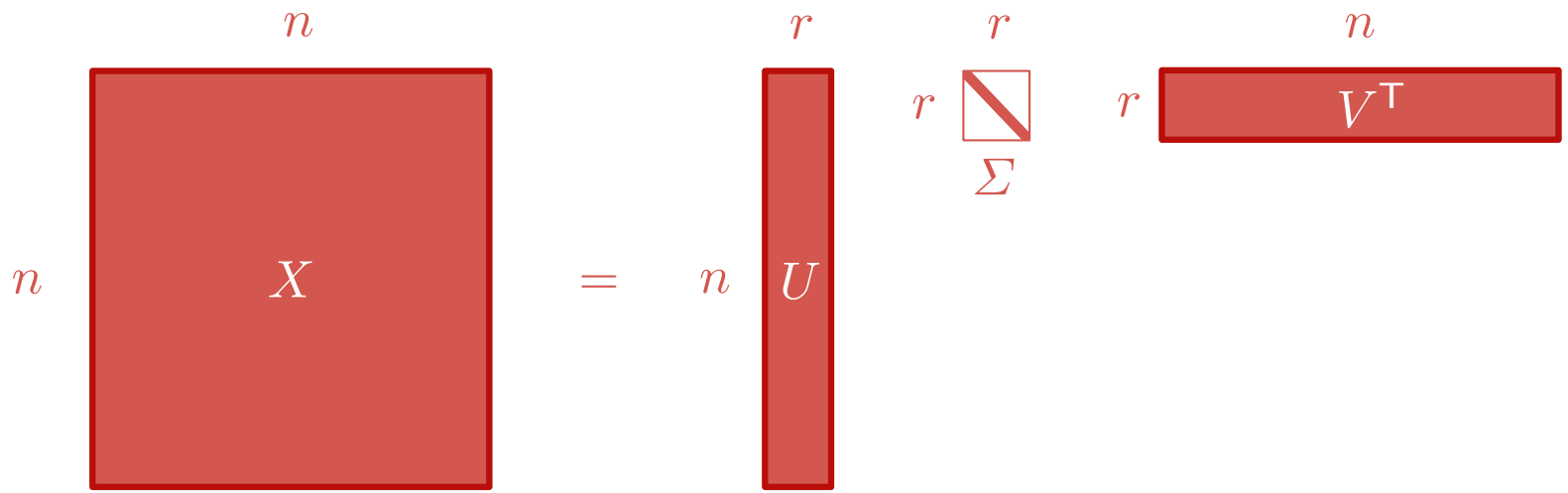
NB: 64 is the number of bits necessary to represent a number in double precision arithmetic.
1 TB = 2^{40} bytes $\approx 10^{12}$ bytes (1 TB is one trillion bytes).

- Since the number of coefficients scales exponentially by d but the accuracy is typically determined by n, this poses a limitation on the size of the problems $\rightsquigarrow$ *Curse of dimensionality*.

Matrix factorization

- One of the possible workarounds $\rightsquigarrow$ use matrix factorization!
  $\rightsquigarrow$ the singular value decomposition (SVD):

$$X = U\Sigma V^\top : \; U^\top U = I_r, \; V^\top V = I_r, \; \Sigma = \operatorname{diag}(\sigma_i), \; \sigma_1 \geqslant \cdots \geqslant \sigma_r > 0.$$



- Only $2nr + r$ coefficients instead of n^2. If $r \ll n$, then big memory savings.
- Perform the calculations directly in the factorized format.

Riemannian trust-region (RTR) method

Algorithm 1: Riemannian trust-region (RTR)

Given $\bar{\Delta} > 0$, $\Delta_1 \in (0, \bar{\Delta})$
for $i = 1, 2, \ldots$ **do**
  Define the second-order model

$$m_i : \mathrm{T}_{x_i}\mathcal{M} \to \mathbb{R},\ \xi \mapsto f(x_i) + \langle \operatorname{grad} f(x_i), \xi \rangle + \frac{1}{2} \langle \operatorname{Hess} f(x_i)[\xi], \xi \rangle.$$

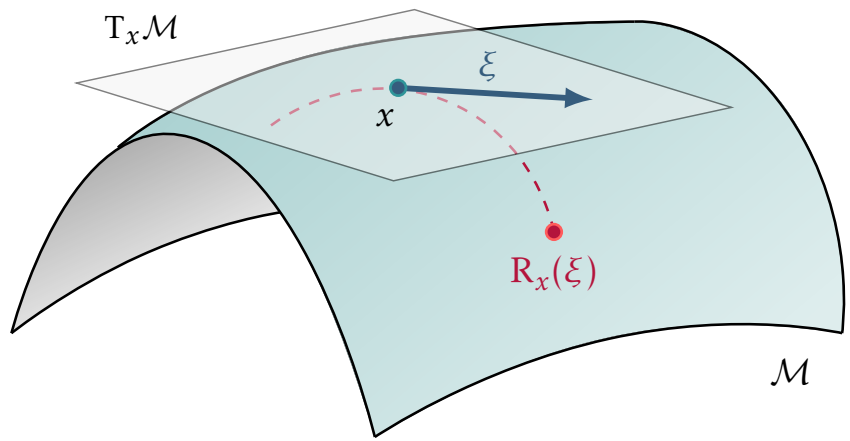
  Trust-region subproblem: compute η_i by solving

$$\eta_i = \operatorname{argmin} m_i(\xi) \quad \text{s.t.} \quad \|\xi\| \le \Delta_i.$$

  Compute $\rho_i = (\widehat{f}(0) - \widehat{f}_i(\eta_i))/(m_i(0) - m_i(\eta_i))$.
  if $\rho_i \ge 0.05$ **then**
    Accept step and set $x_{i+1} = \mathrm{R}_{x_i}(\eta_i)$.
  else
    Reject step and set $x_{i+1} = x_i$.
  end if
  Radius update: set

$$\Delta_{i+1} = \begin{cases} \min(2\Delta_i, \bar{\Delta}) & \text{if } \rho_i \ge 0.75 \text{ and } \|\eta_i\| = \Delta_i, \\ 0.25\,\|\eta_i\| & \text{if } \rho_i \le 0.25, \\ \Delta_i & \text{otherwise.} \end{cases}$$

end for



TR method: [Goldfeld/Quandt/Trotter 1966, Sorensen 1982, Fletcher 1980/1987 ...]
RTR method: [Absil/Baker/Gallivan 2007]

Riemannian Hessian and preconditioning

- In the case of Riemannian submanifolds, the full Riemannian Hessian of f at $x \in \mathcal{M}$ is given by the projected Euclidean Hessian plus the curvature part

$$\operatorname{Hess} f(x)[\xi] = P_x \nabla^2 f(x) P_x + P_x(\text{“curvature terms”}) P_x.$$

  $\rightsquigarrow$ Use $P_x \nabla^2 f(x) P_x$ to develop a preconditioner for the **trust-region subproblem** in the RTR method. This is a symmetric n^2-by-n^2 matrix.

- Inverse of $P_x \nabla^2 f(x) P_x \rightsquigarrow$ good candidate for a preconditioner.

  ⚠ Not inverted directly, since this would cost $\mathcal{O}(n^6)$.

- A good preconditioner should reduce the number of iterations of the inner trust-region solver. It has to be effective and cheap to compute.

  $\rightsquigarrow$ Many (tedious) calculations, but the numerical results are quite striking!

Symmetric positive semidefinite matrices with fixed rank: [Vandereycken/Vandewalle 2010]

The Allen–Cahn equation/1

- Reaction-diffusion equation that models the process of phase separation in multi-component alloy systems.
  - Other applications include mean curvature flows, two-phase incompressible fluids, complex dynamics of dendritic growth, and image segmentation.
- In its simplest form, it reads

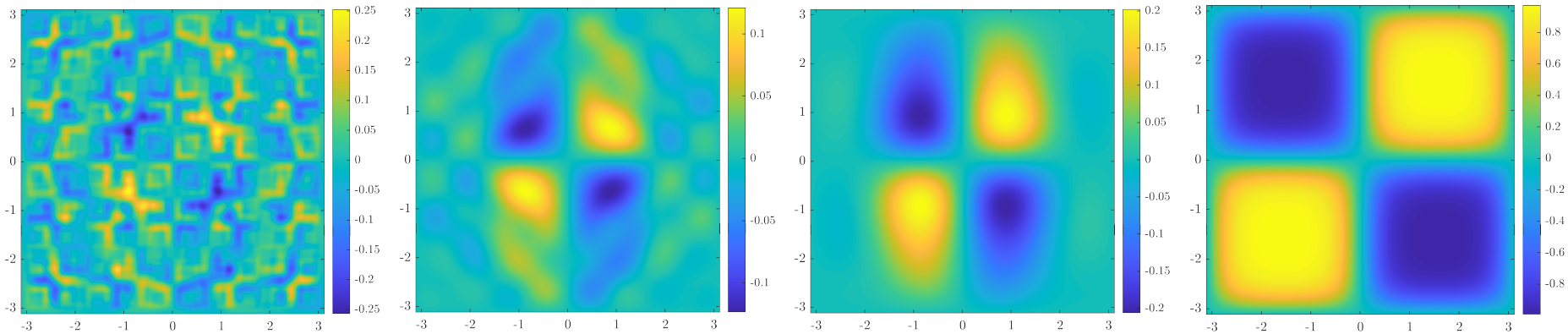
$$\frac{\partial w}{\partial t} = \varepsilon \Delta w + w - w^3,$$

  where $w \equiv w(\boldsymbol{x}, t)$, $\boldsymbol{x} \in \Omega = [-\pi, \pi]^2$, and $t \geq 0$.
- It is a stiff PDE with a low-order polynomial nonlinearity and a diffusion term $\varepsilon \Delta w$.

Allen–Cahn equation: [Allen/Cahn 1972, Allen/Cahn 1973]

The Allen–Cahn equation/2 - "naive" discretization

- Spatial discretization on a uniform grid, 256×256 grid points.
  - Storage of each matrix: $256^2 \times 8/2^{20} \approx 0.5$ MB.
  - The Laplacian Δw is discretized using central finite differences with periodic boundary conditions.
- Numerical time integration with a fourth-order Runge–Kutta method (ERK4), $h = 10^{-4}$, because of the condition for an explicit scheme to be stable (very similar to the CFL condition). Very small time step!



(a) $t = 0$ (b) $t = 0.5$ (c) $t = 2$ (f) $t = 15$

Figure: Time evolution of the solution w to the Allen–Cahn equation, with ERK4, $h = 10^{-4}$.

MATLAB code available at https://github.com/MarcoSutti/PrecRTR

The Allen–Cahn equation/3 - rank assessment

- Question: is it low rank? Preliminary study on the dense-format solution.

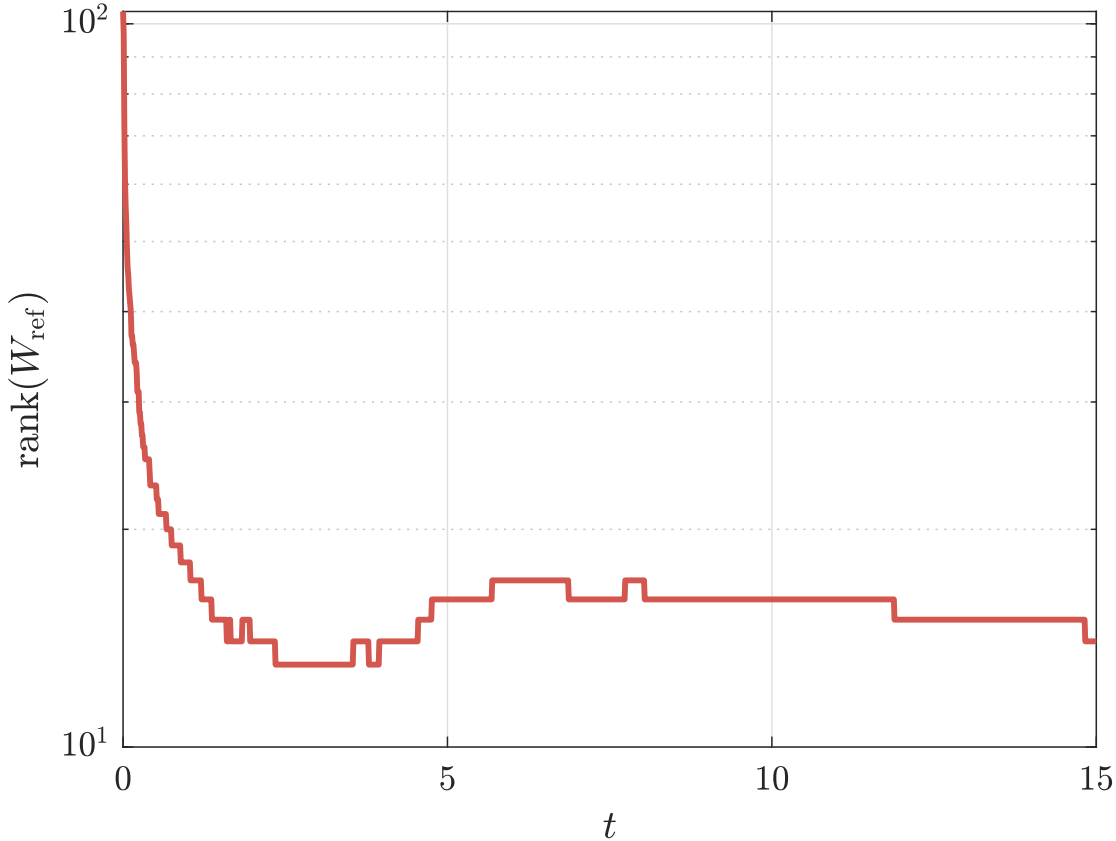


Figure: Time evolution of the rank of the numerical solution W_{ref} to the Allen–Cahn equation, with ERK4, $h = 10^{-4}$.

Implicit numerical integration scheme & low-rank format

- The reference solution in the previous slides is computed with an explicit fourth-order Runge–Kutta method (ERK4), $h = 10^{-4}$. Very small!

  $\rightsquigarrow$ Time to perform the entire simulation until $t = 15$: $\approx$ 36.5 minutes!

- We could use an implicit numerical integration scheme for the time integration.

  ⊕ It allows for a larger time step than its explicit counterpart.

  ⊖ Typically requires the solution of nonlinear equations, which is very expensive.

$\rightsquigarrow$ Idea: Using the low-rank format and the preconditioner to reduce the computational cost together with an implicit numerical time integration scheme that avoids the restriction on the time step due to the stability condition!

The Allen–Cahn equation/4 - low-rank evolution/I

- We build the objective functional

$$\mathcal{F}(w) := \int_{\Omega} \frac{\varepsilon h}{2} \|\nabla w\|^2 + \frac{(1-h)}{2} w^2 + \frac{h}{4} w^4 - \widetilde{w} \cdot w \, \mathrm{d}x \, \mathrm{d}y.$$

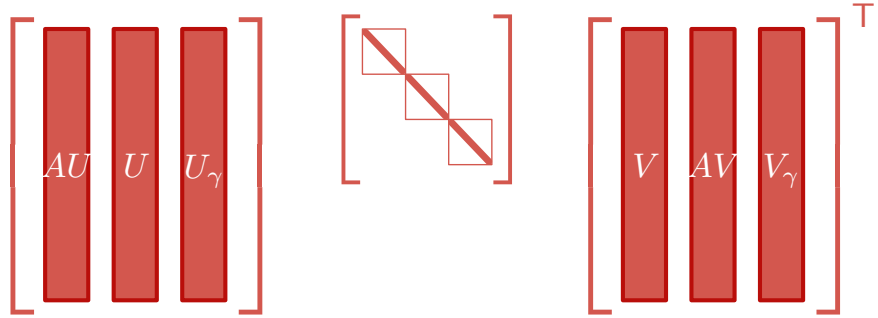
- Discretization of the objective functional in factorized matrix form is

$$F = h_x^2 \left(\frac{\varepsilon h}{2} \left(\|(LU)\Sigma\|_{\mathrm{F}}^2 + \|(LV)\Sigma\|_{\mathrm{F}}^2 \right) + \frac{1-h}{2} \|\Sigma\|_{\mathrm{F}}^2 + \frac{h}{4} \sum_{i,j} w_{ij}^4 - \mathrm{Tr}\left((\widetilde{G}^\top G)(V^\top \widetilde{V}) \right) \right).$$

- RTR is applied to the (discretized) optimization problem

$$\min_{W \in \mathcal{M}_r} F(W).$$

- Factorized form of the Euclidean gradient $G = U_G \Sigma_G V_G^\top$.



The Allen–Cahn equation/5 - low-rank evolution/II

- We can take very big time steps, and still, the numerical solution at the final time has an acceptable error with respect to the reference solution.
  - Time to perform the simulation until $t = 15$, with $h = 0.05$: $\approx$ 5 minutes.
  - Time to perform the simulation until $t = 15$, with $h = 1.00$: $\approx$ 12.5 seconds.

$\rightsquigarrow$ Compare with the $\approx$ 36.5 minutes for the dense format!

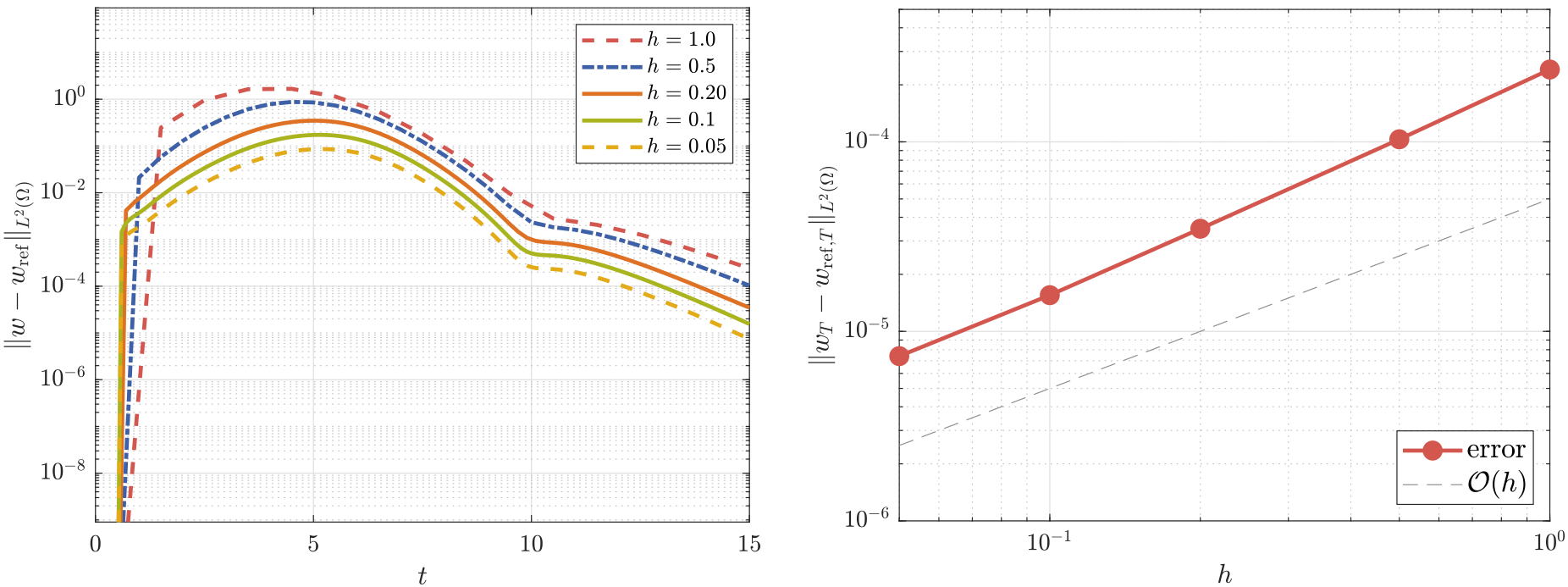


Figure: Panel (a): error versus time for the low-rank evolution of the Allen–Cahn equation. Panel (b): error at $T = 15$ versus time step h.

Implicit low-rank Riemannian schemes - summary

Take-home message: Using a low-rank format allows us to reduce the computational cost and use an implicit numerical time integration scheme that avoids the time step restriction of the stability condition!

Main contributions:

- Efficient preconditioner for the trust-region subproblem on the manifold of fixed-rank matrices.
- Implicit low-rank Riemannian schemes for the time integration of stiff PDEs.

Outlook:

- Use higher-order numerical integration methods.
- Target other applications, e.g., diffusion problems in mathematical biology or problems with low-rank tensor structure.

RGD for spherical area-preserving mappings

Paper: Riemannian gradient descent for spherical area-preserving mappings, M. Sutti and M.-H. Yueh, AIMS Math., Vol. 9(7), 19414–19445, 12 June 2024.

Main contributions:

(i) Combine tools from Riemannian optimization and computational geometry to propose a Riemannian gradient descent (RGD) method for computing spherical area-preserving mappings of topological spheres.

(ii) Numerical experiments on several mesh models demonstrate the accuracy and efficiency of the algorithm.

(iii) Competitiveness and efficiency of our algorithm over three state-of-the-art methods for computing area-preserving mappings.

This part of the talk:

I. Simplicial surfaces and mappings, authalic and stretch energies.

II. Geometry of the power manifold of unit spheres and RGD.

III. Numerical experiments.

Simplicial surfaces and mappings/1

- A simplicial surface $\mathcal{M}$ is the underlying set of a simplicial 2-complex $\mathcal{K}(\mathcal{M}) = \mathcal{F}(\mathcal{M}) \cup \mathcal{E}(\mathcal{M}) \cup \mathcal{V}(\mathcal{M})$ composed of vertices

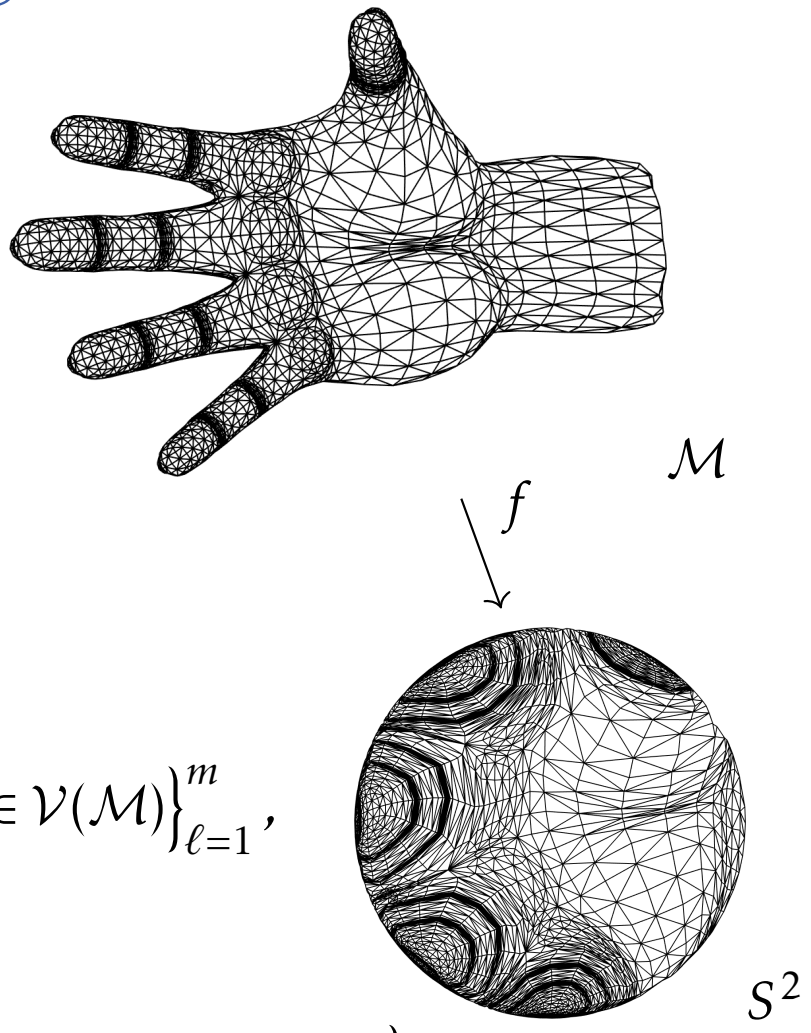
$$\mathcal{V}(\mathcal{M}) = \left\{ v_\ell = \left(v_\ell^1, v_\ell^2, v_\ell^2 \right)^\top \in \mathbb{R}^3 \right\}_{\ell=1}^{n},$$

oriented triangular faces

$$\mathcal{F}(\mathcal{M}) = \left\{ \tau_\ell = [v_{i_\ell}, v_{j_\ell}, v_{k_\ell}] \,|\, v_{i_\ell}, v_{j_\ell}, v_{k_\ell} \in \mathcal{V}(\mathcal{M}) \right\}_{\ell=1}^{m},$$

and undirected edges

$$\mathcal{E}(\mathcal{M}) = \left\{ [v_i, v_j] \,|\, [v_i, v_j, v_k] \in \mathcal{F}(\mathcal{M}) \text{ for some } v_k \in \mathcal{V}(\mathcal{M}) \right\}.$$



- A simplicial mapping $f : \mathcal{M} \to \mathbb{R}^3$ is a particular type of piecewise affine mapping with the restriction mapping $f|_\tau$ being affine, for every $\tau \in \mathcal{F}(\mathcal{M})$.

Simplicial surfaces and mappings/2

- We denote

$$\mathbf{f}_\ell := f(v_\ell) = \left(f_\ell^1, f_\ell^2, f_\ell^3\right)^\top,$$

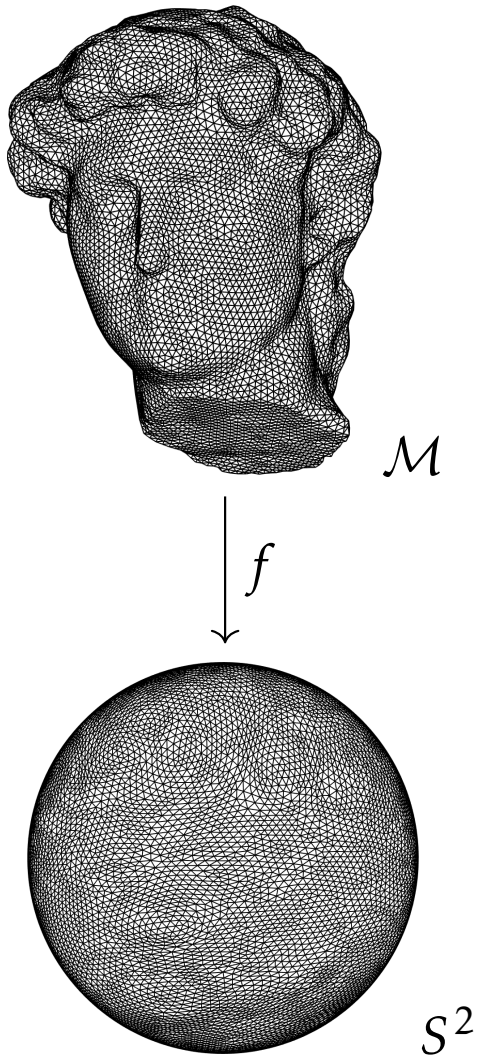
for every $v_\ell \in \mathcal{V}(\mathcal{M})$.

- The (image of the) mapping f can be represented as a matrix

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1^\top \\ \vdots \\ \mathbf{f}_n^\top \end{bmatrix} = \begin{bmatrix} f_1^1 & f_1^2 & f_1^3 \\ \vdots & \vdots & \vdots \\ f_n^1 & f_n^2 & f_n^3 \end{bmatrix} =: \begin{bmatrix} \mathbf{f}^1 & \mathbf{f}^2 & \mathbf{f}^3 \end{bmatrix},$$

or a vector

$$\mathrm{vec}(\mathbf{f}) = \begin{bmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \\ \mathbf{f}^3 \end{bmatrix}.$$



- A simplicial mapping $f : \mathcal{M} \to \mathbb{R}^3$ is said to be area-preserving if $|f(\tau)| = |\tau|$ for every $\tau \in \mathcal{F}(\mathcal{M})$.

Authalic and stretch energies

- The authalic (or equiareal) energy for simplicial mappings $f : \mathcal{M} \to \mathbb{R}^3$ is

$$E_A(f) = E_S(f) - \mathcal{A}(f),$$

where $\mathcal{A}(f)$ is the image area, and E_S is the stretch energy defined as [see Lemma 3.1, Yueh 2023]

$$E_S(f) = \sum_{\tau \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|^2}{|\tau|}.$$

- (If the area-preserving simplicial mapping exists) then every minimizer of $E_S(f)$ is an area-preserving mapping and vice-versa [Theorem 3.3, Yueh 2023], i.e.,

$$f = \operatorname*{argmin}_{|g(\mathcal{M})|=|\mathcal{M}|} E_S(g)$$

if and only if $|f(\tau)| = |\tau|$ for every $\tau \in \mathcal{F}(\mathcal{M})$.

Theoretical foundation of the stretch energy minimization for area-preserving simplicial mappings: [Yueh 2023]

Geometry of the power manifold $\left(S^2\right)^n$

We aim to minimize the function $E(f) = E(\mathbf{f}_1, \ldots, \mathbf{f}_n)$, where each $\mathbf{f}_\ell$, $\ell = 1, \ldots, n$, lives on the same manifold S^2.

$\rightsquigarrow$ This leads us to consider the power manifold of n unit spheres

$$\left(S^2\right)^n = \underbrace{S^2 \times S^2 \times \cdots S^2}_{n \text{ times}},$$

with the metric of S^2 extended elementwise.

Tools from Riemannian geometry needed to use RGD on this manifold:

- The projector onto the tangent space to $\left(S^2\right)^n$ is used to compute the Riemannian gradient.
- The projection onto $\left(S^2\right)^n$ turns points of $\mathbb{R}^{n\times 3}$ into points of $\left(S^2\right)^n$.
- The retraction turns an objective function defined on $\mathbb{R}^{n\times 3}$ into an objective function defined on the manifold $\left(S^2\right)^n$.

Riemannian gradient descent method

Algorithm 2: The RGD method on $\left(S^2\right)^n$.

Given objective function E, power manifold $\left(S^2\right)^n$, initial iterate[(*)] $\mathbf{f}^{(0)} \in \left(S^2\right)^n$, projector $\mathrm{P}_{\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n}$ from $\mathbb{R}^{n\times 3}$ to $\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n$, retraction $\mathrm{R}_{\mathbf{f}}$ from $\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n$ to $\left(S^2\right)^n$;

Result: Sequence of iterates $\{f^{(k)}\}$.

$k \leftarrow 0$;

while $f^{(k)}$ *does not sufficiently minimizes* E **do**

  Compute the Euclidean gradient of the objective function $\nabla E(f^{(k)})$;

  Compute the Riemannian gradient as $\operatorname{grad} E(f^{(k)}) = \mathrm{P}_{\mathrm{T}_{\mathbf{f}^{(k)}}\left(S^2\right)^n}\left(\nabla E(f^{(k)})\right)$;

  Choose the anti-gradient direction $\mathbf{d}^{(k)} = -\operatorname{grad} E(f^{(k)})$;

  Use a line-search procedure to compute a step size $\alpha_k > 0$ that satisfies the sufficient decrease condition;

  Set $\mathbf{f}^{(k+1)} = \mathrm{R}_{\mathbf{f}^{(k)}}(\alpha_k \mathbf{d}^{(k)})$;

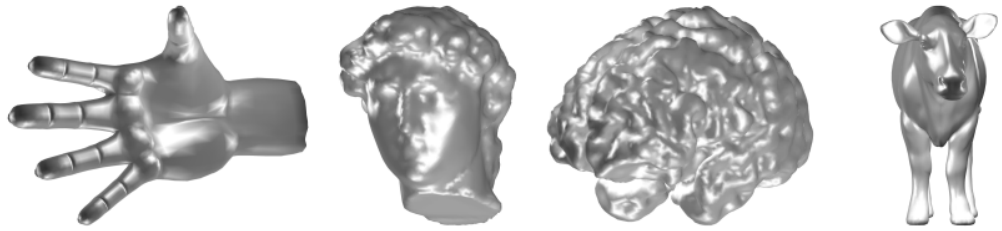
  $k \leftarrow k+1$;

end while

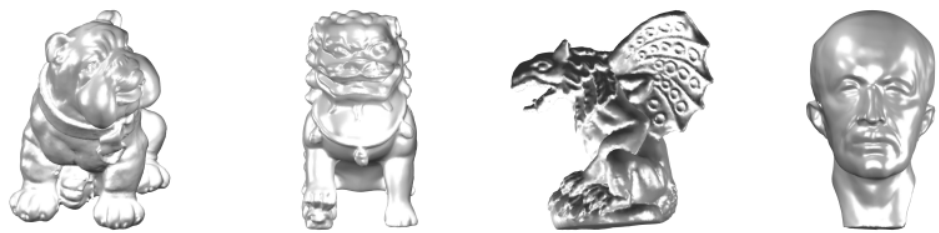
(*) The initial mapping $\mathbf{f}^{(0)} \in \left(S^2\right)^n$ is computed via the fixed-point iteration (FPI) method of Yueh et al., 2019, until the first increase in energy is detected.

The benchmark triangular mesh models

Model Name	Right Hand	David Head	Cortical Surface	Bull
# Faces	8,808	21,338	30,000	34,504
# Vertices	4,406	10,671	15,002	17,254

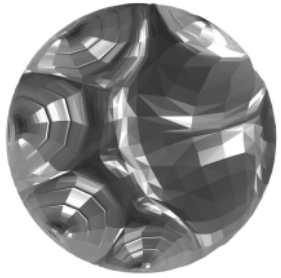
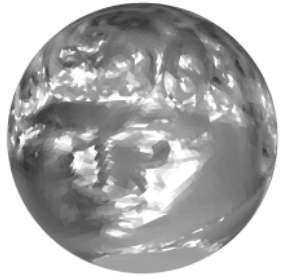


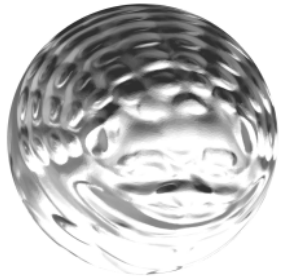
Model Name	Bulldog	Lion Statue	Gargoyle	Max Planck
# Faces	99,590	100,000	100,000	102,212
# Vertices	49,797	50,002	50,002	51,108

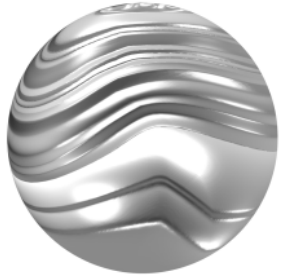


Model Name	Bunny	Chess King	Art Statuette	Bimba
# Faces	111,364	263,712	895,274	1,005,146
# Vertices	55,684	131,858	447,639	502,575

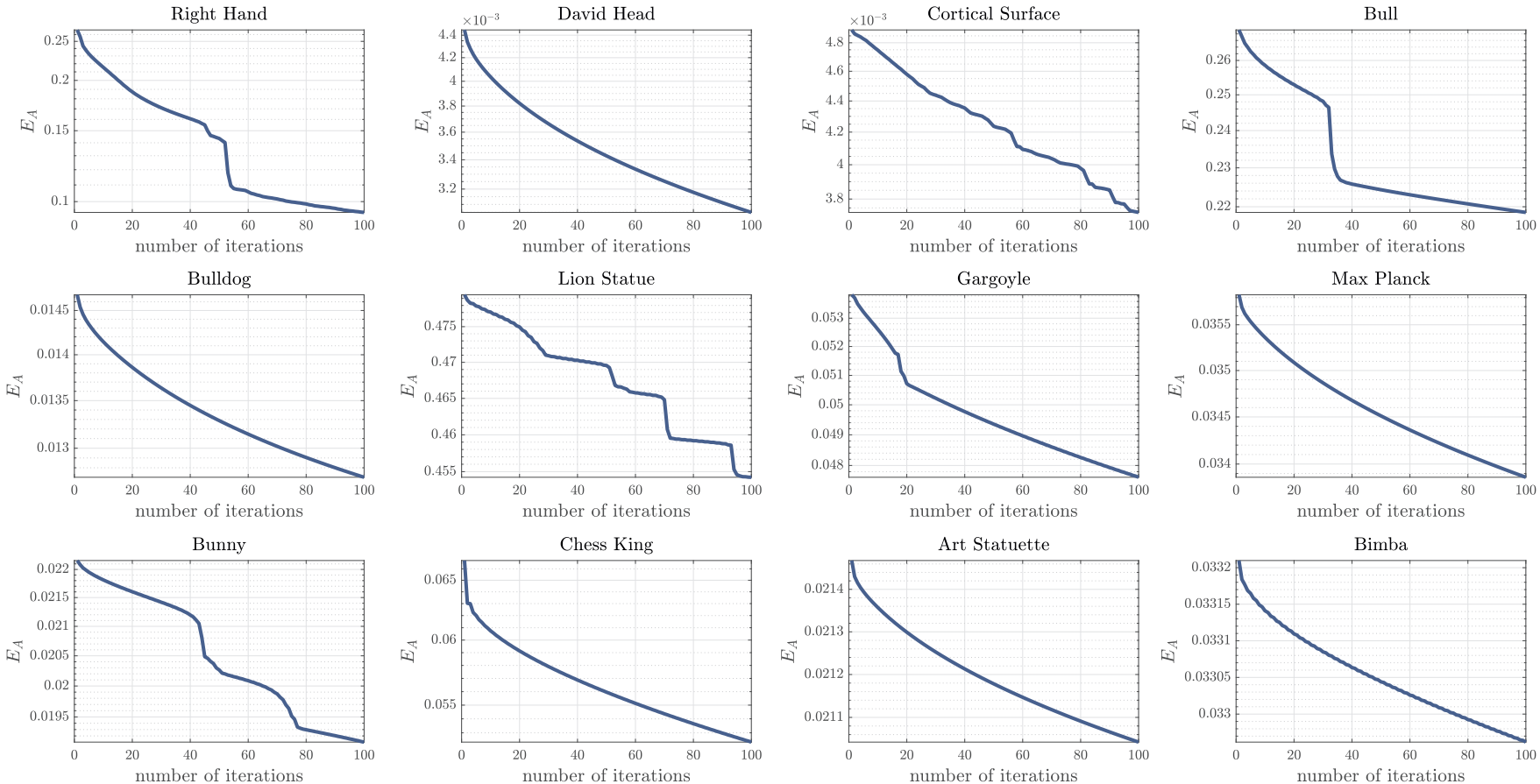
Resulting spherical mappings

Right Hand	David Head	Cortical Surface	Bull
			

Bulldog	Lion Statue	Gargoyle	Max Planck
			

Bunny	Chess King	Art Statuette	Bimba
			

Convergence behavior of RGD



Comparison with other methods

Comparison with the adaptive area-preserving parameterization for genus-zero closed surfaces proposed by Choi et al., 2022.

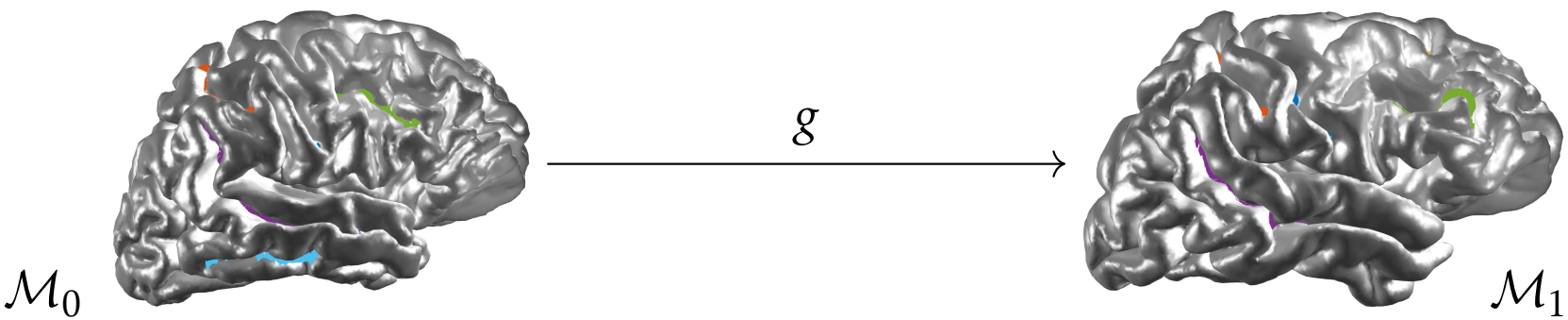
	Choi et al., 2022			**Our RGD method**		
Model Name	SD/Mean	$E_A(f)$	Time	**SD/Mean**	$\boldsymbol{E_A(f)}$	**Time**
Right Hand	18.3283	4.84×10^{3}	218.03	**0.1204**	$\mathbf{9.40 \times 10^{-2}}$	**4.07**
David Head	0.0426	2.27×10^{-2}	298.76	**0.0156**	$\mathbf{3.04 \times 10^{-3}}$	**9.16**
Cortical Surface	0.6320	1.14×10^{0}	420.20	**0.0200**	$\mathbf{3.72 \times 10^{-3}}$	**16.01**
Bull	8.5565	1.82×10^{3}	34.42	**0.1348**	$\mathbf{2.19 \times 10^{-1}}$	**18.89**
Bulldog	9.2379	1.22×10^{3}	183.94	**0.0343**	$\mathbf{1.27 \times 10^{-2}}$	**61.93**
Lion Statue	0.2626	8.96×10^{-1}	1498.91	**0.1894**	$\mathbf{4.54 \times 10^{-1}}$	**76.76**
Gargoyle	0.3558	1.30×10^{0}	1483.35	**0.0646**	$\mathbf{4.76 \times 10^{-2}}$	**80.52**
Max Planck	11.6875	1.49×10^{3}	195.39	**0.0525**	$\mathbf{3.39 \times 10^{-2}}$	**75.60**
Bunny	27.6014	8.94×10^{3}	157.87	**0.0390**	$\mathbf{1.91 \times 10^{-2}}$	**89.62**
Chess King	11.8300	1.65×10^{3}	608.55	**0.0647**	$\mathbf{5.23 \times 10^{-2}}$	**207.47**
Art Statuette	394.4414	9.93×10^{0}	2284.79	**0.0405**	$\mathbf{2.10 \times 10^{-2}}$	**654.57**
Bimba Statue	0.5110	2.01×10^{0}	16 773.34	**0.0512**	$\mathbf{3.29 \times 10^{-2}}$	**775.36**

Adaptive area-preserving parameterization for genus-zero closed surfaces: [Choi/Giri/Kumar 2022]

Surface registration

- A registration mapping between surfaces $\mathcal{M}_0$ and $\mathcal{M}_1$ is a bijective mapping $g\colon \mathcal{M}_0 \to \mathcal{M}_1$. An ideal registration mapping keeps important landmarks aligned while preserving specified geometry properties.
- Framework for the computation of landmark-aligned area-preserving parameterizations of genus-zero closed surfaces.
- Illustration with the landmark-aligned morphing process from one brain to another.

Problem statement: Given a set of landmark pairs $\{(p_i, q_i) \mid p_i \in \mathcal{M}_0,\ q_i \in \mathcal{M}_1\}_{i=1}^{m}$, our goal is to compute an area-preserving simplicial mapping $g\colon \mathcal{M}_0 \to \mathcal{M}_1$ that satisfies $g(p_i) \approx q_i$, for $i = 1, \ldots, m$.

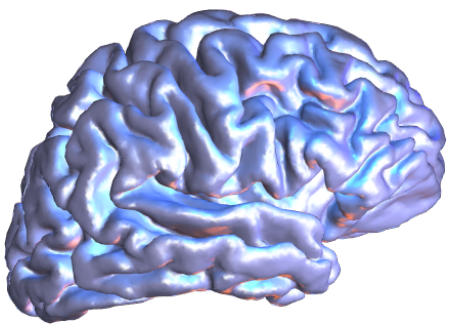


Brain morphing

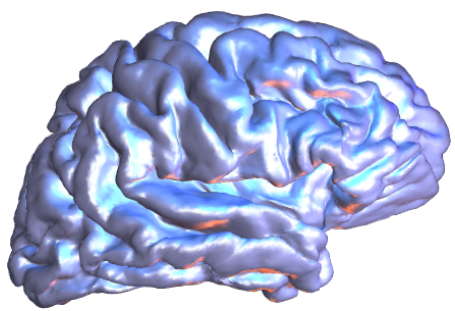
- Brain morphing via the linear homotopy method.
- A landmark-aligned morphing process from $\mathcal{M}_0$ to $\mathcal{M}_1$ can be constructed by the linear homotopy $H\colon \mathcal{M}_0 \times [0,1] \to \mathbb{R}^3$ defined as

$$H(v,t) = (1-t)\,v + t\,g(v).$$

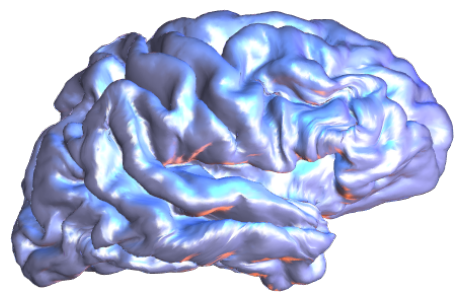
- We demonstrate the morphing process from one brain to another brain by four snapshots at four different values of t.



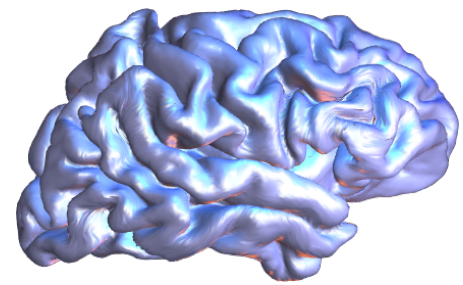
$H(\mathcal{M}_0, 0) = \mathcal{M}_0$



$H(\mathcal{M}_0, 0.33)$



$H(\mathcal{M}_0, 0.67)$



$H(\mathcal{M}_0, 1) = \mathcal{M}_1$

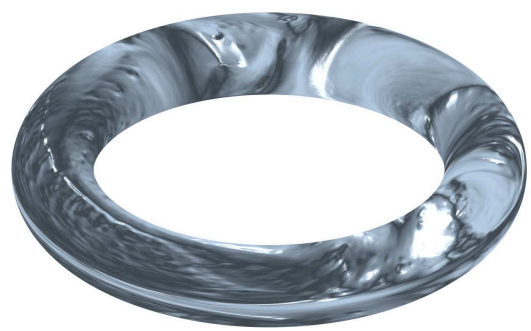
RGD for area-preserving mappings - summary

Main contributions:

- Riemannian optimization & computational geometry $\rightsquigarrow$ RGD method for computing spherical area-preserving mappings of genus-zero closed surfaces.
- Extensive numerical experiments on various mesh models to demonstrate the algorithm's stability and effectiveness.
- Landmark-aligned surface registration between two human brain models.

Outlook and ongoing work:

- Enhance the speed of convergence of the algorithm using appropriate Riemannian generalizations of the conjugate gradient method or the limited memory BFGS method.
- Target area-preserving mappings of genus-one closed surfaces, e.g., the ring torus. Ongoing work with Mei-Heng Yueh.

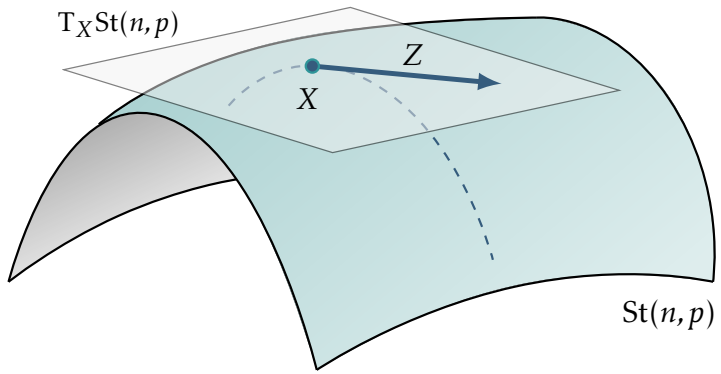


Shooting methods for computing geodesics on Stiefel

Paper: A single shooting method with approximate Fréchet derivative for computing geodesics on the Stiefel manifold, M. Sutti, Electron. Trans. Numer. Anal. (ETNA), Vol. 60, 501–519, September 2024.

- Many applications in diverse fields deal with data belonging to the Stiefel manifold

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n\times p} : \ X^\top X = I_p\}.$$



- Evaluation of the distance between two points on $\mathrm{St}(n,p)$.
- No closed-form solution is known for $\mathrm{St}(n,p)$!

This part of the talk:

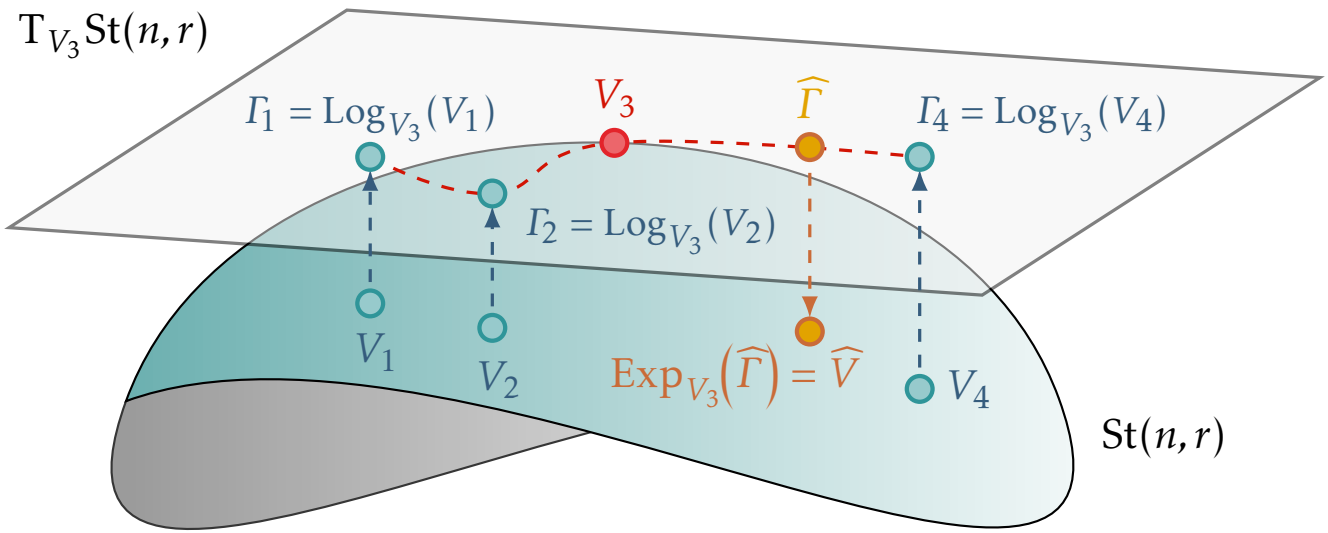
I. Motivating example: interpolation on manifolds.

II. Computational framework based on the shooting method.

III. Comparisons with other state-of-the-art methods.

A motivating example: interpolation on manifolds

- Model order reduction (MOR) for dynamical systems parametrized according to $p = [p_1, \ldots, p_d]^\top$.
- Suppose we have a set of local orthonormal basis matrices $\{V_1, V_2, \ldots, V_K\}$.
- Given a new parameter value $\hat{p}$, a basis $\widehat{V}$ can be obtained by **interpolating** the local basis matrices on a tangent space to $\mathrm{St}(n, r)$.
- For interpolation on $\mathrm{T}_{V_3}\mathrm{St}(n, r)$, the distance is needed.

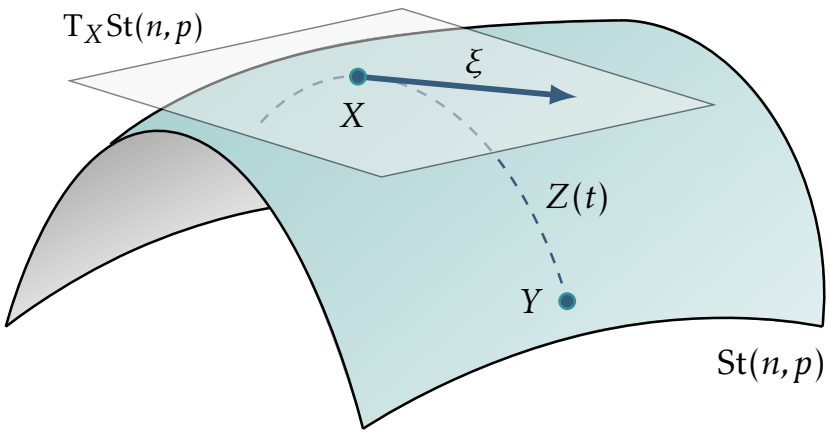


MOR, POD: [Benner/Gugercin/Willcox 2015]
Interpolation on manifolds: [Hüper/Silva Leite 2007, Amsallem 2010, Amsallem/Farhat 2011]

Riemannian distance on $\mathrm{St}(n,p)$

- Property: Given $X, Y \in \mathrm{St}(n,p)$, s.t. $\mathrm{Exp}_X(\xi) = Y$, the Riemannian distance $d(X,Y)$ equals the length of $\xi \equiv \dot{Z}(0) \in \mathrm{T}_X\mathrm{St}(n,p)$:

$$d(X,Y) = \|\xi\|_c = \sqrt{\langle \xi, \xi \rangle_c}.$$



Equivalent to: Compute the length of the Riemannian logarithm of Y with base point X, i.e.,

$$\mathrm{Log}_X(Y) = \xi.$$

- No closed-form solution is known for $\mathrm{St}(n,p)$!

$\rightsquigarrow$ How do we compute $d(X,Y)$ in practice / numerically?

Single shooting for BVPs

- Boundary value problem (BVP): Find $w(x)\colon [a,b] \to \mathbb{R}$ that satisfies

$$w'' = f(x, w, w'), \quad \text{with BCs} \quad \begin{cases} w(a) = \alpha, \\ w(b) = \beta. \end{cases}$$

- Recast it as an initial value problem (IVP): Find $w(x)$ that satisfies

$$w'' = f(x, w, w'), \quad \text{with ICs} \quad \begin{cases} w(a) = \alpha, \\ w'(a) = s. \end{cases}$$

$\rightsquigarrow$ In general, this has a unique solution $w(x) \equiv w(x; s)$ which depends on s (Picard–Lindelöf theorem). Analytical or numerical solution (e.g., Runge–Kutta).

$\rightsquigarrow$ Single shooting method for BVPs:

- Define $F(s) = w(b; s) - \beta$.
- Find $\bar{s}$ s.t. $F(\bar{s}) = 0$. Usually, with Newton's method.

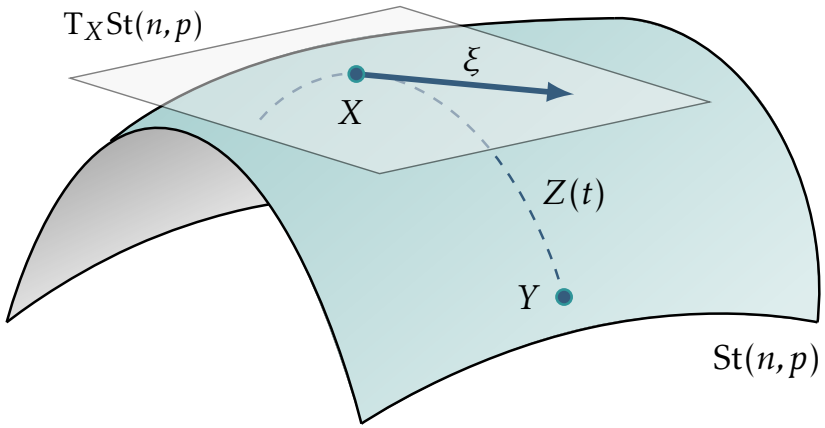
BVPs and shooting methods: see, e.g., [Stoer/Bulirsch 1991]

Stiefel geodesics via single shooting

- Problem statement:
  Find $\xi \equiv \dot{Z}(0) \in \mathrm{T}_X\mathrm{St}(n,p)$
  that satisfies the BVP

$$\ddot{Z} = -\dot{Z}\dot{Z}^\top Z - Z((Z^\top \dot{Z})^2 + \dot{Z}^\top \dot{Z}),$$

$$\text{with BCs} \quad \begin{cases} Z(0) = X, \\ Z(1) = Y. \end{cases}$$



- Recall: we have the explicit solution: $Z(t) = [X\ X_\perp] \exp\left(\begin{bmatrix} X^\top\xi & -(X_\perp^\top\xi)^\top \\ X_\perp^\top\xi & O \end{bmatrix} t\right) \begin{bmatrix} I_p \\ O \end{bmatrix}.$

$\rightsquigarrow$ Single shooting for Stiefel geodesics:

- Define $F(\xi) = Z_{(t=1,\xi)} - Y$.
- Find ξ s.t. $F(\xi) = 0$ with Newton's method.

Derivation and approximation/1

- Starting from the nonlinear equation

$$F(\xi) = Z_{(t=1,\xi)} - Y,$$

use matrix perturbation theory

$$F(\xi + \delta\xi) = Z_1(\xi + \delta\xi) - Y_1 = 0,$$

$$Z_1(\xi + \delta\xi) = Z_1(\xi) + \mathrm{D}Z_1\,[\delta\xi^{(k)}] + o(\|\delta\xi\|).$$

- Perturbation of the matrix exponential by a matrix $E \in \mathbb{R}^{n\times n}$ is

$$\exp_{\mathrm{m}}(A + E) = \exp_{\mathrm{m}}(A) + \mathrm{D}\exp_{\mathrm{m}}(A)[E] + o(\|E\|).$$

- Representation for the Fréchet derivative of the matrix exponential

$$\mathrm{D}\exp_{\mathrm{m}}(A)[E] = E + \frac{AE + EA}{2} + \frac{A^2E + AEA + EA^2}{3!} + \cdots.$$

- Approximation: keep only the first two terms in the expansion, i.e.,

$$\mathrm{D}\exp_{\mathrm{m}}(A)[E] \approx E + \frac{AE + EA}{2}.$$

Function of matrices: [Higham 2008]

Derivation and approximation/2

- The last formula can be used to approximate $\mathrm{D}\exp_{\mathrm{m}}(A(\xi))\Big[\mathrm{D}A(\xi)[\delta\xi]\Big]$ in $\mathrm{D}Z_1\,[\delta\xi^{(k)}]$

$$Z_1(\xi) + \mathrm{D}Z_1\,[\delta\xi^{(k)}] - Y_1 = 0,$$

  resulting in

$$Q \cdot \Big(\mathrm{D}A(\xi)[\delta\xi] + \tfrac{1}{2}(A \cdot \mathrm{D}A(\xi)[\delta\xi] + \mathrm{D}A(\xi)[\delta\xi] \cdot A)\Big) \cdot I_{n,p} = Y_1 - Z_1.$$

- This is now a linear matrix equation to be solved for $\delta\xi$.
- We obtain a (small-sized) Sylvester equation which can be efficiently solved with MATLAB's command `lyap` to obtain the update $\delta\xi$.

Algorithm pseudocode

Algorithm 3: A single shooting method on the Stiefel manifold with an approximation of the Fréchet derivative (SSAF method).

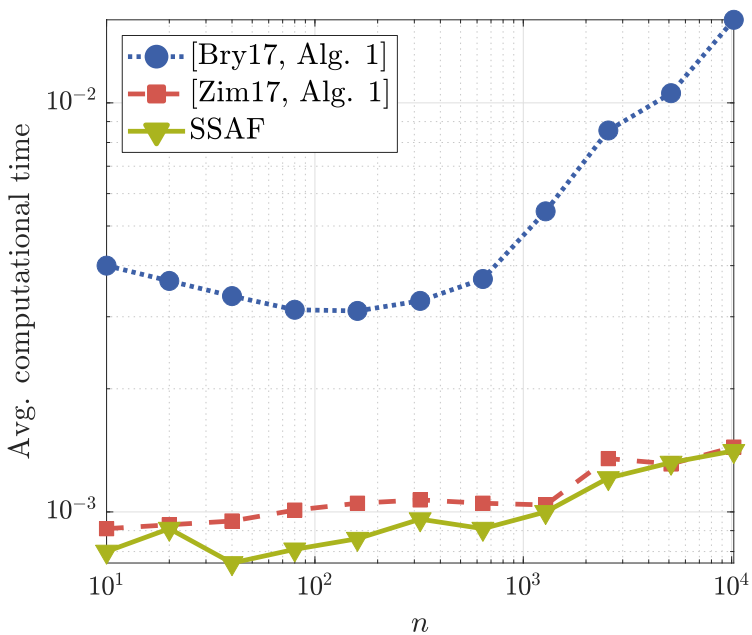
Given Y_0, Y_1;
Result: ξ^* such that $\mathrm{Exp}_{Y_0}(\xi^*) = Y_1$.
Compute the initial guess $\xi^{(0)}$;
Set $k = 0$;
while *a stopping criterion is met* **do**
    Compute $F^{(k)} = Z_1(1, \xi^{(k)}) - Y_1$;
    Solve $F^{(k)} + \mathrm{D}Z_1\,[\delta\xi^{(k)}] = 0$ for $\delta\xi^{(k)}$;
    Update $\xi^{(k+1)} \leftarrow \xi^{(k)} + \delta\xi^{(k)}$;
    Project $\xi^{(k+1)}$ onto $\mathrm{T}_{Y_0}\mathrm{St}(n,p)$: $\xi^{(k+1)} \leftarrow \mathrm{P}_{Y_0}\left(\xi^{(k+1)}\right)$;
    $k = k + 1$;
end while

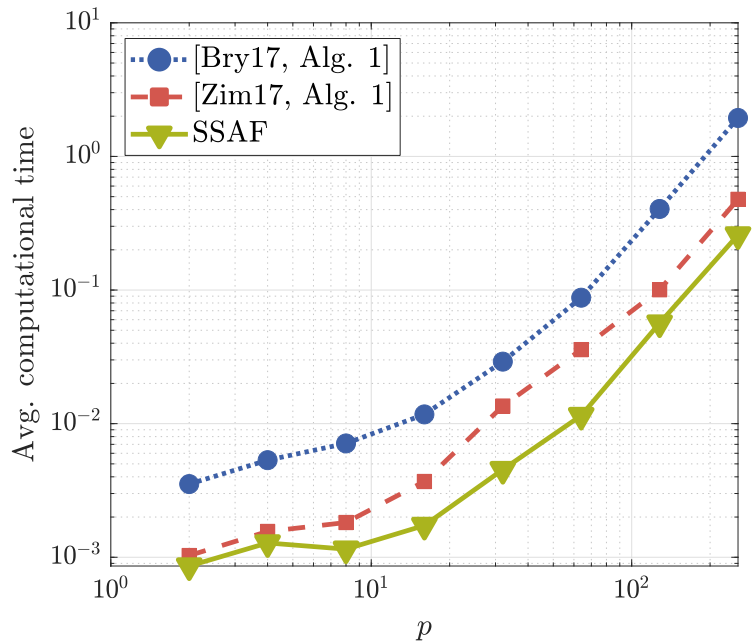
- The code was implemented in MATLAB and is freely available on the repository https://github.com/MarcoSutti/SSAF_2024_repo

Comparisons with other methods

Table: Comparisons on $\mathrm{St}(1500, p)$ with large values of p, for a prescribed $d(X, Y) = 0.5\,\pi$. Results are averaged over 10 pairs of randomly generated endpoints on $\mathrm{St}(1500, p)$.

p	Avg. comput. time (s)			Avg. no. of iterations		
	Bryner, 2017	Zimmermann, 2017	**SSAF, 2024**	Bryner, 2017	Zimmermann, 2017	**SSAF, 2024**
500	12.09	3.30	**1.89**	3.00	2.00	**4.00**
700	31.27	8.21	**4.56**	3.00	2.00	**4.00**
1000	77.37	20.39	**8.82**	3.00	2.00	**4.00**





Shooting for computing geodesics on Stiefel - summary

Main contributions:

- Computational framework with the shooting method to compute the Riemannian distance on the Stiefel manifold.
- Computational trick: approximation of the Fréchet derivative involved.
- Competitiveness and efficiency of our SSAF method w.r.t. the state-of-the-art algorithms.

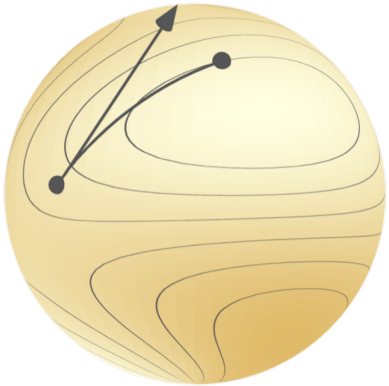
Outlook and ongoing work:

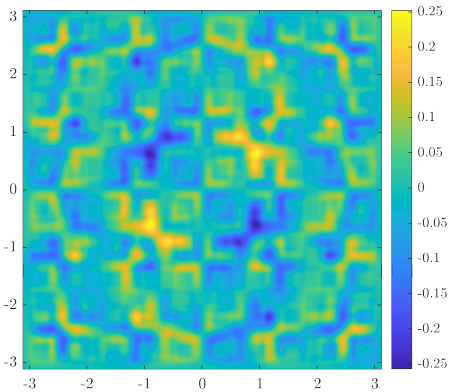
- Nonlinear Schwarz methods to compute geodesics on manifolds. Joint work with Tommaso Vanzan. Tech. report (submitted), September 2024. arXiv preprint arXiv:2409.01023.

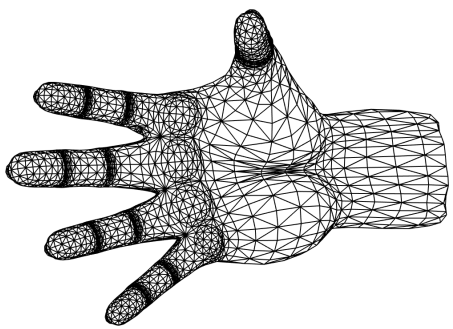
Optimization on matrix manifolds - summary

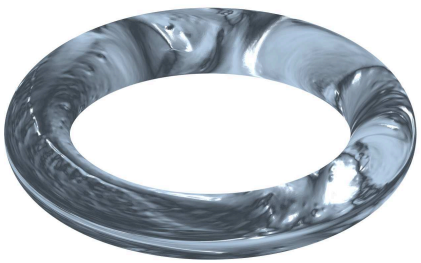
This talk:

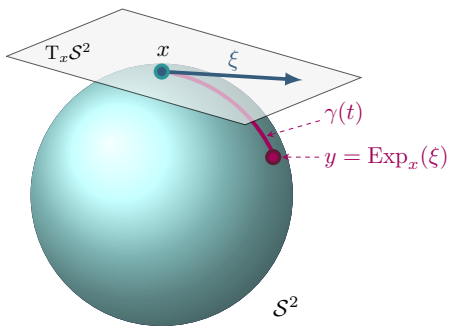
- Riemannian optimization framework.
  - Solid theoretical foundation.
  - Algorithmic components derived from Riemannian geometry.
- A framework for a wide range of applications:
  - Data compression, computer graphics, computing geodesics reduced order models…
- Common key idea: exploit structure to create novel and efficient algorithms.











Thank you for your attention!

III. Bonus material

Line-search (LS) method

$\rightsquigarrow$ **How to calculate t_k?**

- **Exact** line search (LS):

$$\min_{t \geq 0} f(x_k + t\eta_k)$$

  - t_k^{EX} is the unique minimizer if f is strictly convex.
  - Can sometimes be computed. Good for theory.

- In practice, for generic f, we do not use exact LS. Replace exact LS with something computationally cheaper, but still effective.

  $\rightsquigarrow$ **Armijo line-search** (also known as Armijo backtracking, Armijo condition, sufficient decrease condition, ...).

Armijo line-search technique: [Armijo 1966]

Retractions on embedded submanifolds

Let $\mathcal{M}$ be an embedded submanifold of a vector space $\mathcal{E}$. Thus $\mathrm{T}_x\mathcal{M}$ is a linear subspace of $\mathrm{T}_x\mathcal{E} \simeq \mathcal{E}$. Since $x \in \mathcal{M} \subseteq \mathcal{E}$ and $\xi \in \mathrm{T}_x\mathcal{M} \subseteq \mathrm{T}_x\mathcal{E} \simeq \mathcal{E}$, with little abuse of notation we write $x + \xi \in \mathcal{E}$.

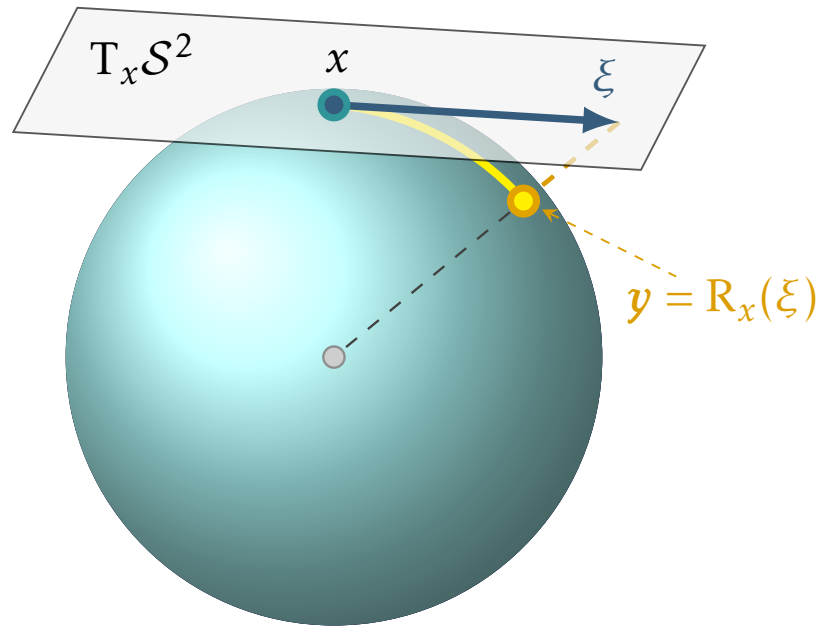
$\rightsquigarrow$ General recipe to define a retraction $\mathrm{R}_x(\xi)$ for embedded submanifolds:

- Move along ξ to get to $x + \xi$ in $\mathcal{E}$.
- Map $x + \xi$ back to $\mathcal{M}$. For matrix manifolds, use matrix decompositions.

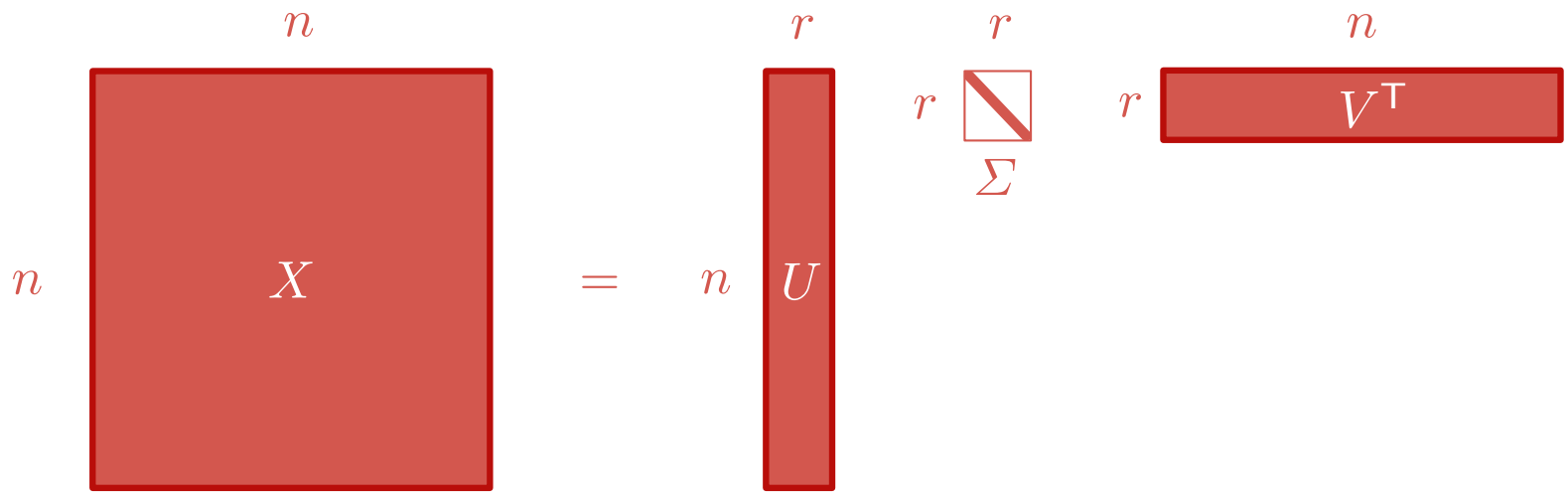
Example. Let $\mathcal{M} = S^{n-1}$, then the retraction at $x \in S^{n-1}$ is

$$\mathrm{R}_x(\xi) = \frac{x + \xi}{\|x + \xi\|},$$

defined for all $\xi \in \mathrm{T}_x S^{n-1}$. $\mathrm{R}_x(\xi)$ is the point on S^{n-1} that minimizes the distance to $x + \xi$.



Motivation for the low-rank format/2



- Storing a dense 5000×5000 matrix in double precision takes $5000^2 \times 8/2^{20} \approx 191$ MB.
  - If it has rank 10 and we store only its factors, it takes $(2 \times 5000 \times 10 + 10) \times 8/2^{20} = 0.76$ kB.
  - If it has rank 100 and we store only its factors, it takes $(2 \times 5000 \times 100 + 100) \times 8/2^{20} = 7.63$ MB.
- For a matrix stored in the dense format, the storage complexity grows as n^2, but if the matrix is stored in low-rank format, then the storage grows as nr.

Riemannian Hessian and preconditioning/2

- Applying the preconditioner in $X \in \mathcal{M}_r$ means solving for $\xi \in \mathrm{T}_X\mathcal{M}$ the system

$$H_X \operatorname{vec}(\xi) = \operatorname{vec}(\eta),$$

where $\eta \in \mathrm{T}_X\mathcal{M}$ is a known tangent vector.

- This is equivalent to

$$\mathrm{P}_X(A\xi + \xi A) = \eta.$$

- Using the definition of the orthogonal projector onto $\mathrm{T}_X\mathcal{M}_r$, we obtain

$$P_U(A\xi + \xi A)P_V + P_U^{\perp}(A\xi + \xi A)P_V + P_U(A\xi + \xi A)P_V^{\perp} = \eta,$$

which is equivalent to the system

$$\begin{cases} P_U(A\xi + \xi A)P_V = P_U \eta P_V, \\ P_U^{\perp}(A\xi + \xi A)P_V = P_U^{\perp} \eta P_V, \\ P_U(A\xi + \xi A)P_V^{\perp} = P_U \eta P_V^{\perp}. \\ \vdots \end{cases}$$

$\rightsquigarrow$ Many (tedious) calculations, but the numerical results are quite striking!

"LYAP" variational problem

Table: Effect of preconditioning: dependence on size for LYAP.

		Rank 5						Rank 10					
Prec.	size	10	11	12	13	14	15	10	11	12	13	14	15
No	n_{outer}	51	54	61	59	162	92	300	103	61	63	62	59
	$\sum n_{\text{inner}}$	4561	9431	21066	36556	30069	30096	27867	30025	33818	45760	44467	38392
	$\max n_{\text{inner}}$	1801	3191	7055	9404	1194	1851	2974	3385	8894	24367	24537	25013
Yes	n_{outer}	41	45	50	52	56	60	44	64	62	53	56	56
	$\sum n_{\text{inner}}$	44	45	50	52	56	60	69	104	82	60	69	56
	$\max n_{\text{inner}}$	4	1	1	1	1	1	9	9	8	8	8	1

- Stopping criterion: maximum number of outer iterations $n_{\max\ \text{outer}} = 300$. The inner solver is stopped when $\sum n_{\text{inner}}$ first exceeds 30 000.
- Impressive reduction in the number of iterations of the inner solver.
- n_{outer} and $\sum n_{\text{inner}}$ depend (quite mildly) on size, while $\max n_{\text{inner}}$ is basically constant.

An example of factorized gradient

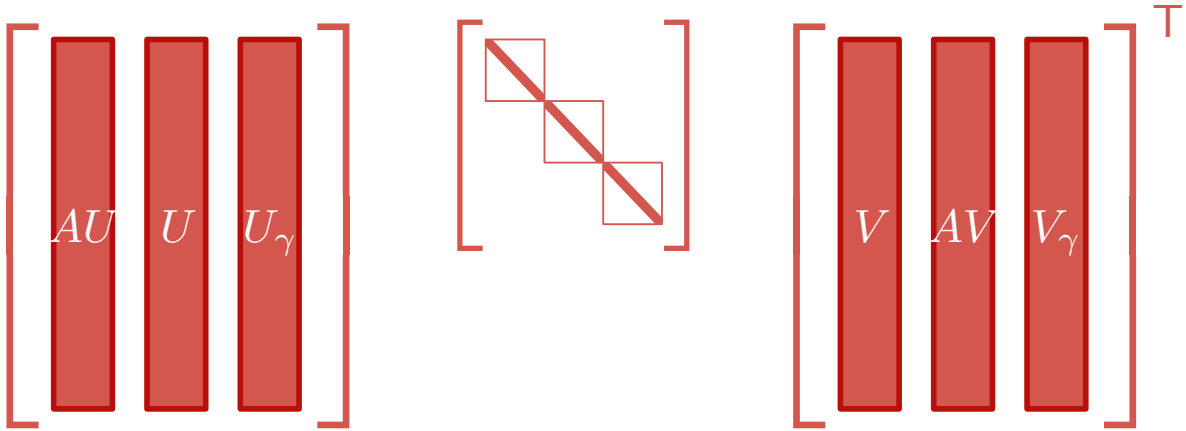
- "LYAP" functional: $\mathcal{F}(w(x,y)) = \int_\Omega \frac{1}{2}\|\nabla w(x,y)\|^2 - \gamma(x,y)\, w(x,y)\,\mathrm{d}x\,\mathrm{d}y$.
- The gradient of $\mathcal{F}$ is the variational derivative $\frac{\delta \mathcal{F}}{\delta w} = -\Delta w - \gamma$.
- The discretized Euclidean gradient in matrix form is given by

$$G = AW + WA - \Gamma.$$

  with A is the second-order periodic finite difference differentiation matrix.
- The first-order optimality condition $G = AW + WA - \Gamma = 0$ is a Lyapunov (or Sylvester) equation.

$\rightsquigarrow$ Factorized Euclidean gradient:

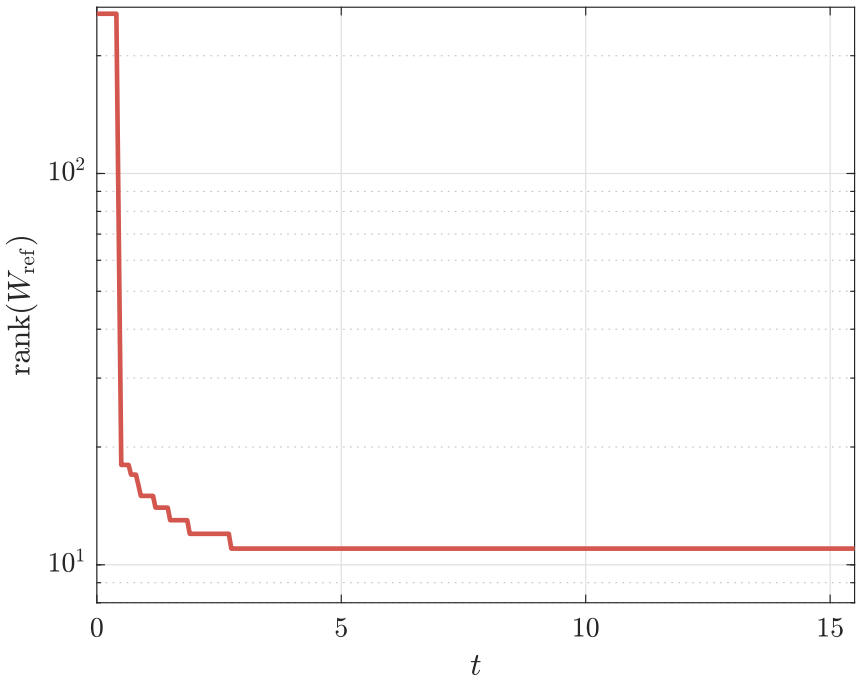
$$G = \begin{bmatrix} AU & U & U_\gamma \end{bmatrix} \operatorname{blkdiag}\left(\Sigma, \Sigma, \Sigma_\gamma\right) \begin{bmatrix} V & AV & V_\gamma \end{bmatrix}^\mathsf{T}.$$



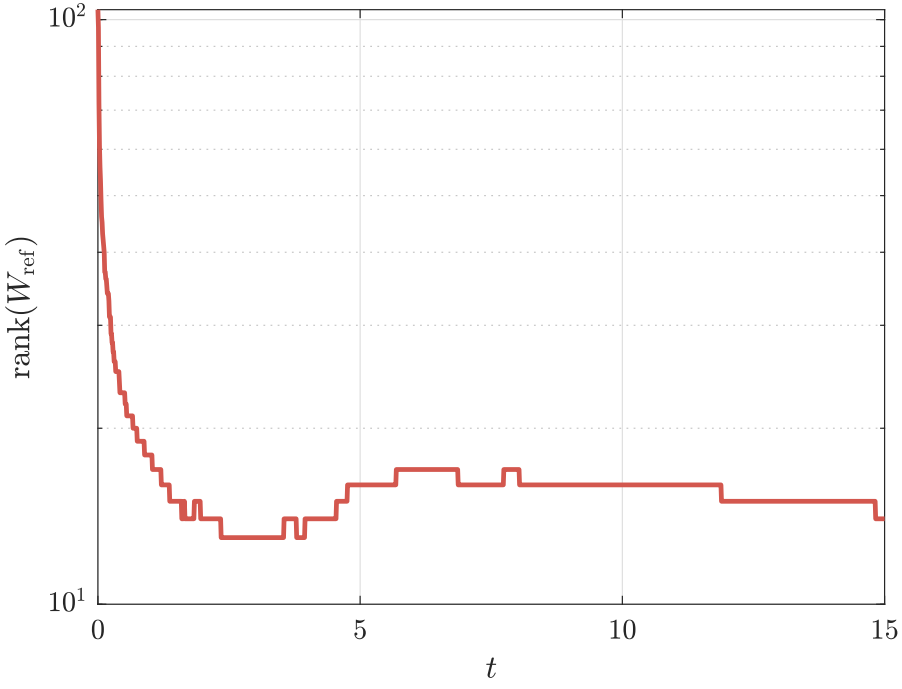
The Allen–Cahn equation/5 - low-rank evolution/II

- We build the functional

$$\min_{w} \mathcal{F}(w) := \int_{\Omega} \frac{\varepsilon h}{2} \|\nabla w\|^2 + \frac{(1-h)}{2} w^2 + \frac{h}{4} w^4 - \widetilde{w} \cdot w \, \mathrm{d}x \, \mathrm{d}y.$$



(a)



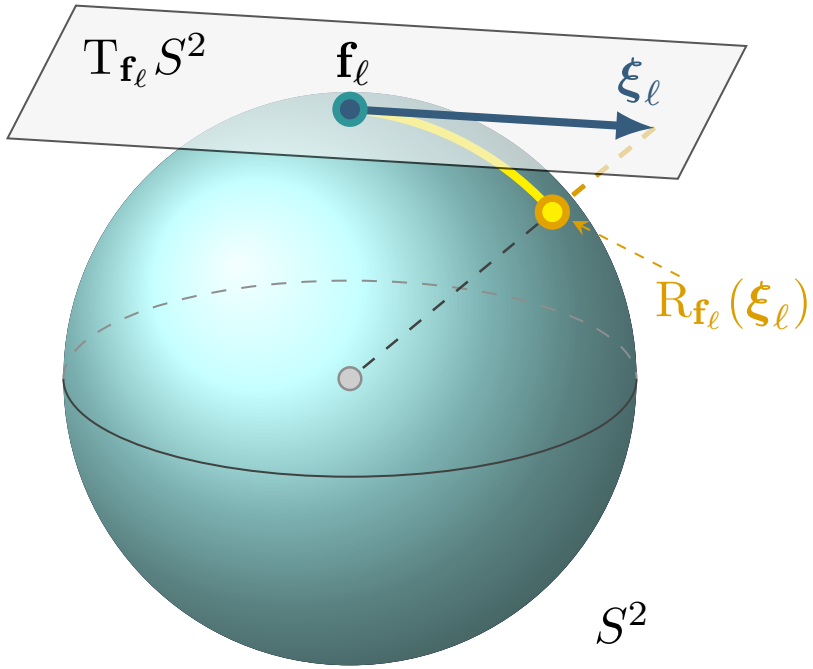
(b)

Figure: Panel (a): error versus time for the low-rank evolution of the Allen–Cahn equation. Panel (b): rank evolution of the reference dense-format solution W_{ref}.

Retraction

- The retraction of a tangent vector $\boldsymbol{\xi}_\ell$ from $\mathrm{T}_{\mathbf{f}_\ell} S^2$ to S^2 is a mapping $\mathrm{R}_{\mathbf{f}_\ell} : \mathrm{T}_{\mathbf{f}_\ell} S^2 \to S^2$, defined by

$$\mathrm{R}_{\mathbf{f}_\ell}(\boldsymbol{\xi}_\ell) = \frac{\mathbf{f}_\ell + \boldsymbol{\xi}_\ell}{\|\mathbf{f}_\ell + \boldsymbol{\xi}_\ell\|}.$$



- For the power manifold $\left(S^2\right)^n$, the retraction of all the tangent vectors $\boldsymbol{\xi}_\ell$, $\ell = 1, \ldots, n$, is a mapping $\mathrm{R}_{\mathbf{f}} : \mathrm{T}_{\mathbf{f}}\left(S^2\right)^n \to \left(S^2\right)^n$, defined by

$$\begin{bmatrix} \boldsymbol{\xi}_1 & \cdots & \boldsymbol{\xi}_n \end{bmatrix}^\top \mapsto \operatorname{diag}\left(\frac{1}{\|\mathbf{f}_1 + \boldsymbol{\xi}_1\|_2}, \ldots, \frac{1}{\|\mathbf{f}_n + \boldsymbol{\xi}_n\|_2} \right) \begin{bmatrix} \mathbf{f}_1 + \boldsymbol{\xi}_1 & \cdots & \mathbf{f}_n + \boldsymbol{\xi}_n \end{bmatrix}^\top.$$

Constructing retractions: [Absil/Malick 2012]

Comparison with other methods/1

Comparison with the fixed-point iteration method for minimizing the authalic energy E_A of Yueh et al., 2019.

	Fixed point method [Yueh et al. 19]			Our RGD method		
Model Name	SD/Mean	$E_A(f)$	Time	SD/Mean	$E_A(f)$	Time
Right Hand	0.4598	2.92×10^{0}	1.35	0.1204	9.40×10^{-2}	4.07
David Head	0.0169	3.58×10^{-3}	4.30	0.0156	3.04×10^{-3}	9.16
Cortical Surface	0.0174	3.21×10^{-3}	5.62	0.0200	3.72×10^{-3}	16.01
Bull	0.1876	4.59×10^{-1}	6.90	0.1348	2.19×10^{-1}	18.89
Bulldog	0.1833	3.99×10^{-1}	22.22	0.0343	1.27×10^{-2}	61.93
Lion Statue	0.2064	5.28×10^{-1}	23.67	0.1894	4.54×10^{-1}	76.76
Gargoyle	4.1020	4.85×10^{2}	36.10	0.0646	4.76×10^{-2}	80.52
Max Planck	0.1844	1.67×10^{1}	25.99	0.0525	3.39×10^{-2}	75.60
Bunny	0.0394	3.96×10^{-2}	35.78	0.0390	1.91×10^{-2}	89.62
Chess King	1.0903	1.79×10^{1}	88.04	0.0647	5.23×10^{-2}	207.47
Art Statuette	0.0908	1.07×10^{-1}	342.95	0.0405	2.10×10^{-2}	654.57
Bimba Statue	0.0932	7.42×10^{-2}	305.00	0.0512	3.29×10^{-2}	775.36

Fixed-point iteration method for minimizing the authalic energy: [Yueh et al. 2019]

Metrics and geodesics on $\mathrm{St}(n,p)$

Embedded metric:

$$\langle \xi, \eta \rangle = \mathrm{Tr}(\xi^\top \eta).$$

Canonical metric:

$$\langle \xi, \eta \rangle_c = \mathrm{Tr}(\xi^\top (I - \tfrac{1}{2} X X^\top) \eta).$$

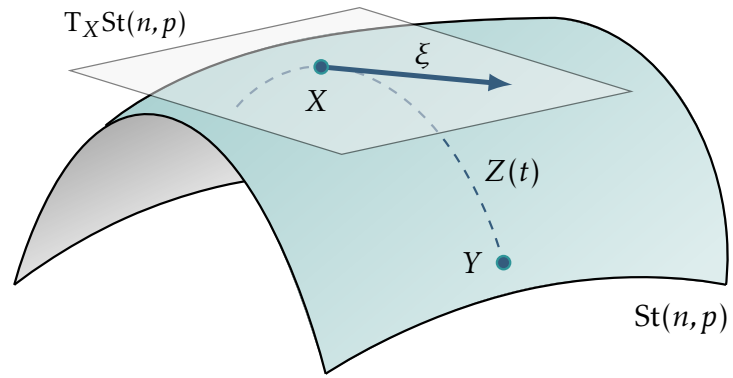
Length of a tangent vector $\xi = X\Omega + X_\perp K$:

$$\|\xi\|_F = \sqrt{\langle \xi, \xi \rangle} = \sqrt{\|\Omega\|_F^2 + \|K\|_F^2}.$$

$$\|\xi\|_c = \sqrt{\langle \xi, \xi \rangle_c} = \sqrt{\tfrac{1}{2}\|\Omega\|_F^2 + \|K\|_F^2}.$$

- Closed-form solution (with the canonical metric) for a geodesic $Z(t)$ that realizes ξ with base point X:

$$Z(t) = [\, X \;\; X_\perp \,] \exp\left(\begin{bmatrix} X^\top \xi & -(X_\perp^\top \xi)^\top \\ X_\perp^\top \xi & O \end{bmatrix} t \right) \begin{bmatrix} I_p \\ O \end{bmatrix}.$$



Riemannian exponential and logarithm

- Given $x \in \mathcal{M}$ and $\xi \in T_x\mathcal{M}$, the exponential mapping $\mathrm{Exp}_x : T_x\mathcal{M} \to \mathcal{M}$ s.t. $\mathrm{Exp}_x(\xi) := \gamma(1)$, with γ being the geodesic with $\gamma(0) = x$, $\dot{\gamma}(0) = \xi$.
- Corollary: $\mathrm{Exp}_x(t\xi) := \gamma(t)$, for $t \in [0, 1]$.
- $\forall x,\ y \in \mathcal{M}$, the mapping $\mathrm{Exp}_x^{-1}(y) \in T_x\mathcal{M}$ is called logarithm mapping.

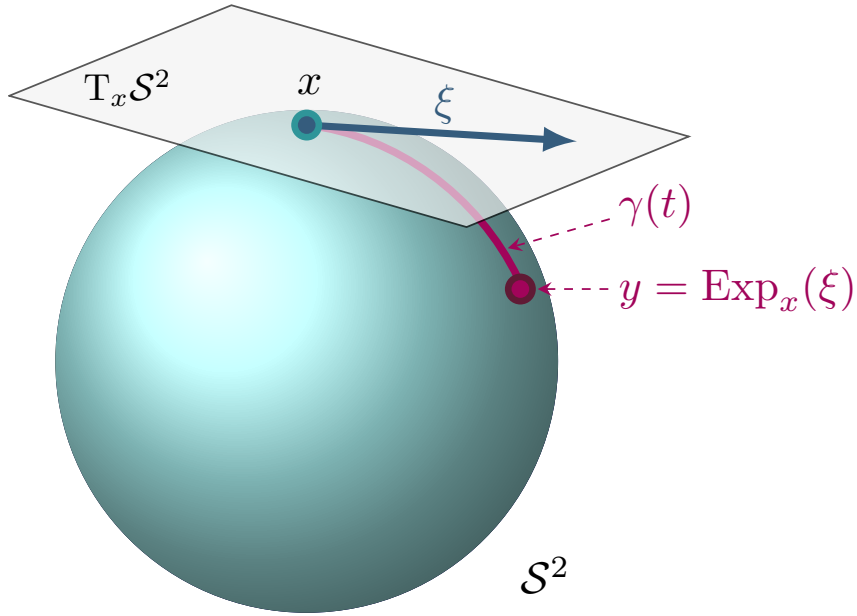
Example. Let $\mathcal{M} = S^{n-1}$, then the exponential mapping at $x \in S^{n-1}$ is

$$y = \mathrm{Exp}_x(\xi) = x\cos(\|\xi\|) + \frac{\xi}{\|\xi\|}\sin(\|\xi\|),$$

and the Riemannian logarithm is

$$\mathrm{Log}_x(y) = \xi = \arccos(x^\top y)\frac{\mathrm{P}_x y}{\|\mathrm{P}_x y\|},$$

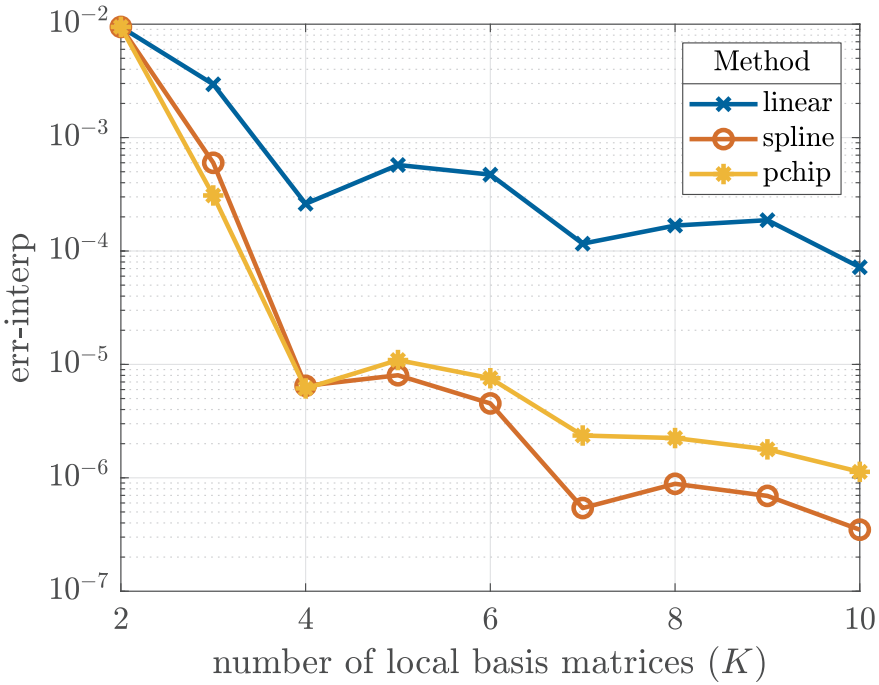
where $y \equiv \gamma(1)$ and P_x is the projector onto $\big(\mathrm{span}(x)\big)^\perp$, i.e., $\mathrm{P}_x = I - xx^\top$.

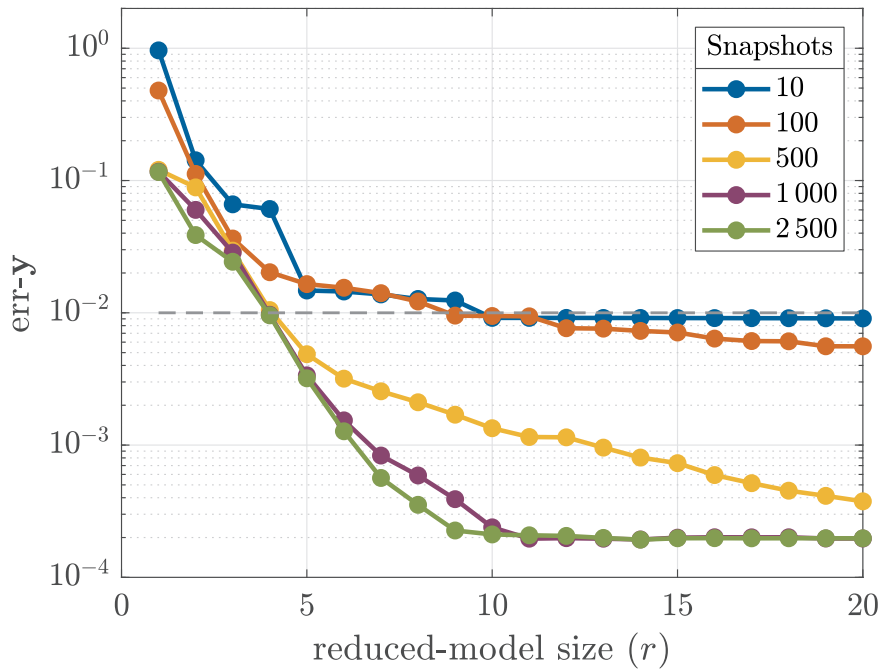


A motivating example: ROM/3

Transient heat equation on a square domain, with 4 disjoint discs.

- FEM discretization with $n = 1169$. Simulation for $t \in [0, 500]$, with $\Delta t = 0.1$.
- 500 snapshot POD over 5000 timeframes, with a reduced model of size $r = 4$.
- Relative error between $y(\cdot; \hat{p})$ and $y_r(\cdot; \hat{p})$ is about 1%.





Details for these experiments: [S. 2023]

Comparisons with other methods/2

Table: Comparisons on $\mathrm{St}(1000, p)$, for doubling values of p, for a prescribed $d(X, Y) = 0.5\,\pi$. $T = 20$, tolerance $\tau = 10^{-5}$. Results are averaged over 100 experiments.

p	Avg. comput. time (s)			Avg. no. of iterations		
	Bryner, 2017	Zimmermann, 2017	SSAF, 2024	Bryner, 2017	Zimmermann, 2017	SSAF, 2024
20	0.03897	0.00641	0.00391	4.00	3.00	5.02
40	0.09512	0.02957	0.01284	3.00	3.00	5.00
80	0.25528	0.08044	0.03969	3.00	2.00	4.00
160	0.76246	0.24119	0.13763	3.00	2.00	4.00
320	3.99810	1.07286	0.64483	3.00	2.00	4.00
640	23.36386	5.62897	2.80133	3.00	2.00	4.00

- The numerical results demonstrate the competitiveness of our SSAF method in terms of both average computation time and number of iterations w.r.t. the existing algorithms considered here.