Optimization on Manifolds

Justin Solomon

6.838: Shape Analysis

Spring 2021

MIT EECS

Common Constraints in ML/Vision

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

- **Euclidean space $\mathbb{R}^n$**
- **Unit sphere S^{n-1}**
- **Stiefel manifold $V_k(\mathbb{R}^n)$**
  orthonormal *k*-frames
- **Grassmann manifold $\mathrm{Gr}(k, V)$**
  k*-dimensional linear subspaces of *V
- **Rotation group $\mathrm{SO}(n)$**
- **Semidefinite matrices S_+^n**

From "Optimizing in smooth waters: optimization on manifolds" (Boumal, 2015)

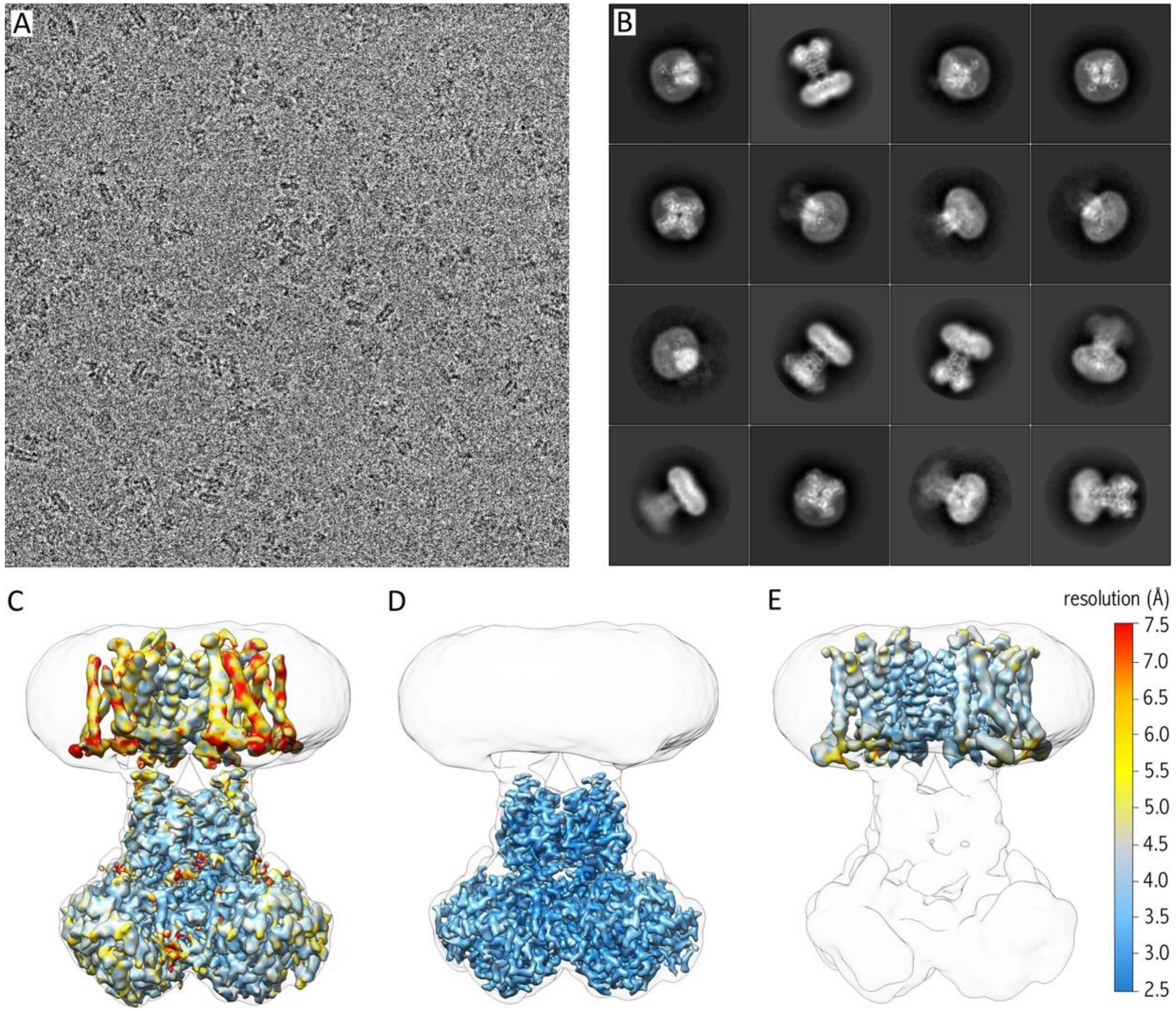
Example: Structure-from-Motion

Search space: set of rotations

SfM (Princeton Vision & Robotics group)

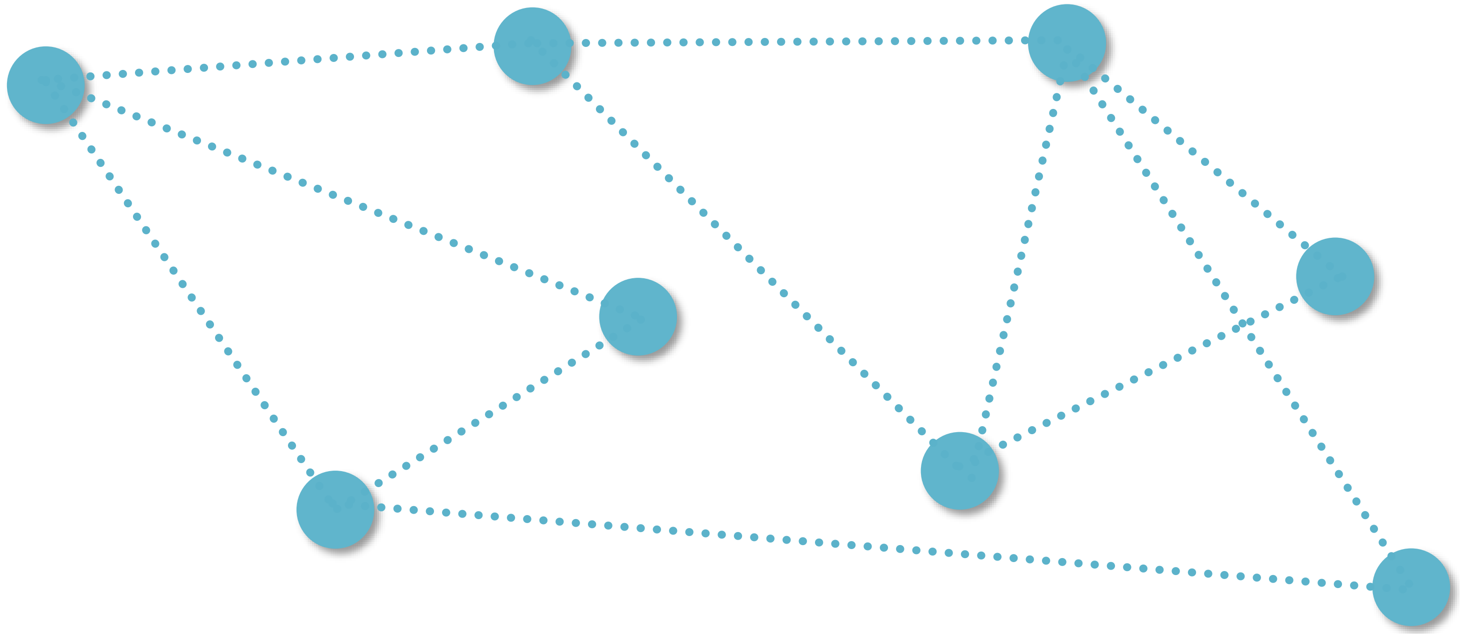
Slide courtesy Nicolas Boumal

Example: Cryo-EM



https://elifesciences.org/articles/37558

Example: Sensor Network Localization



Search space: cloud of points, up to rigid motion

Slide courtesy Nicolas Boumal

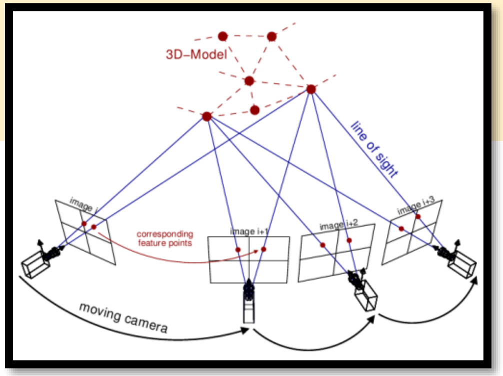
Typical Approach

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

constraint

Example:

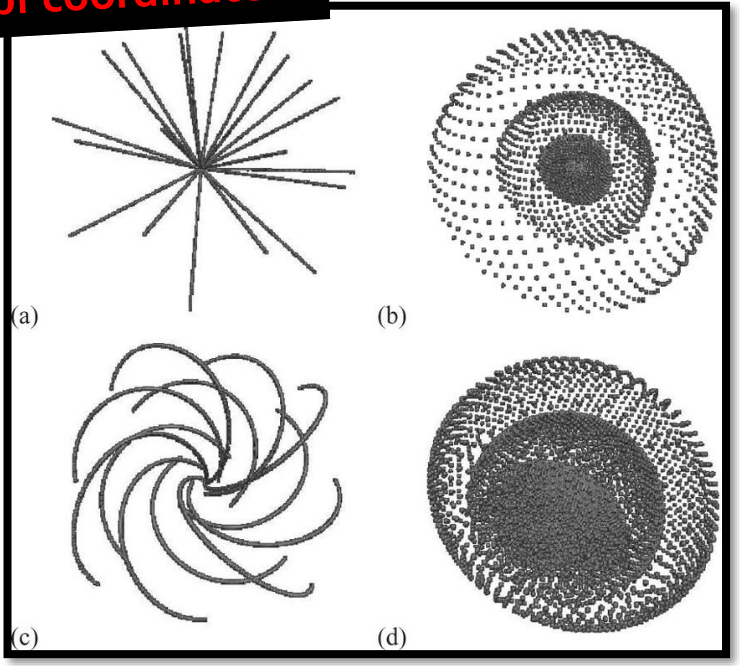
$$\min_{R \in \mathrm{SO}(3)} f(R) \longmapsto \min_{\substack{R^\top R = I_3 \\ \det(R) = 1}} f(R)$$



Challenges

$$\min_{\substack{R^\top R = I_3 \\ \det(R)=1}} f(R)$$

"Hopf coordinates"

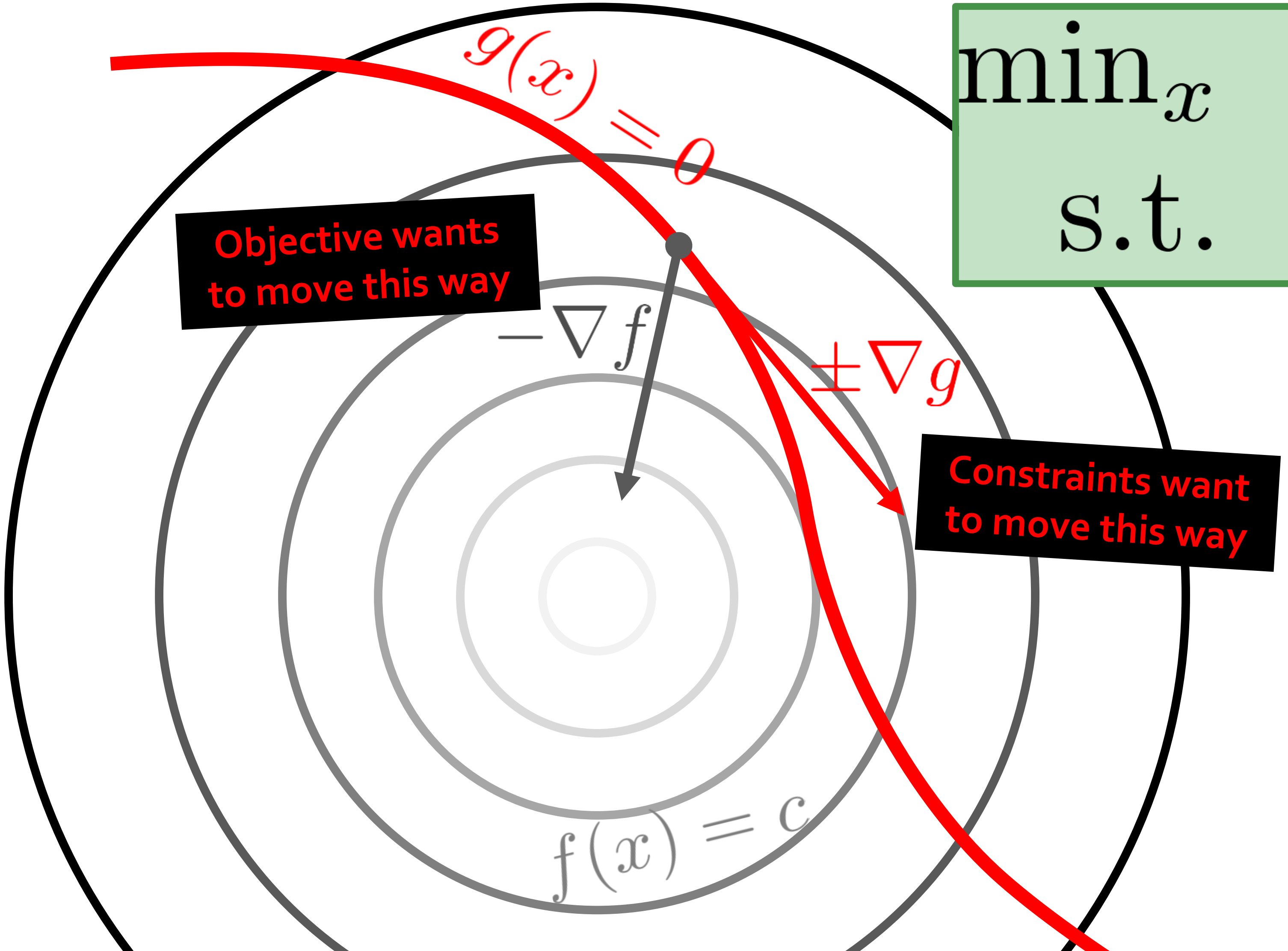


Constraints cut out a **nonconvex, curved set** **in $\mathbb{R}^{3\times 3}$**

Naively: Degree 2 & 3 polynomials

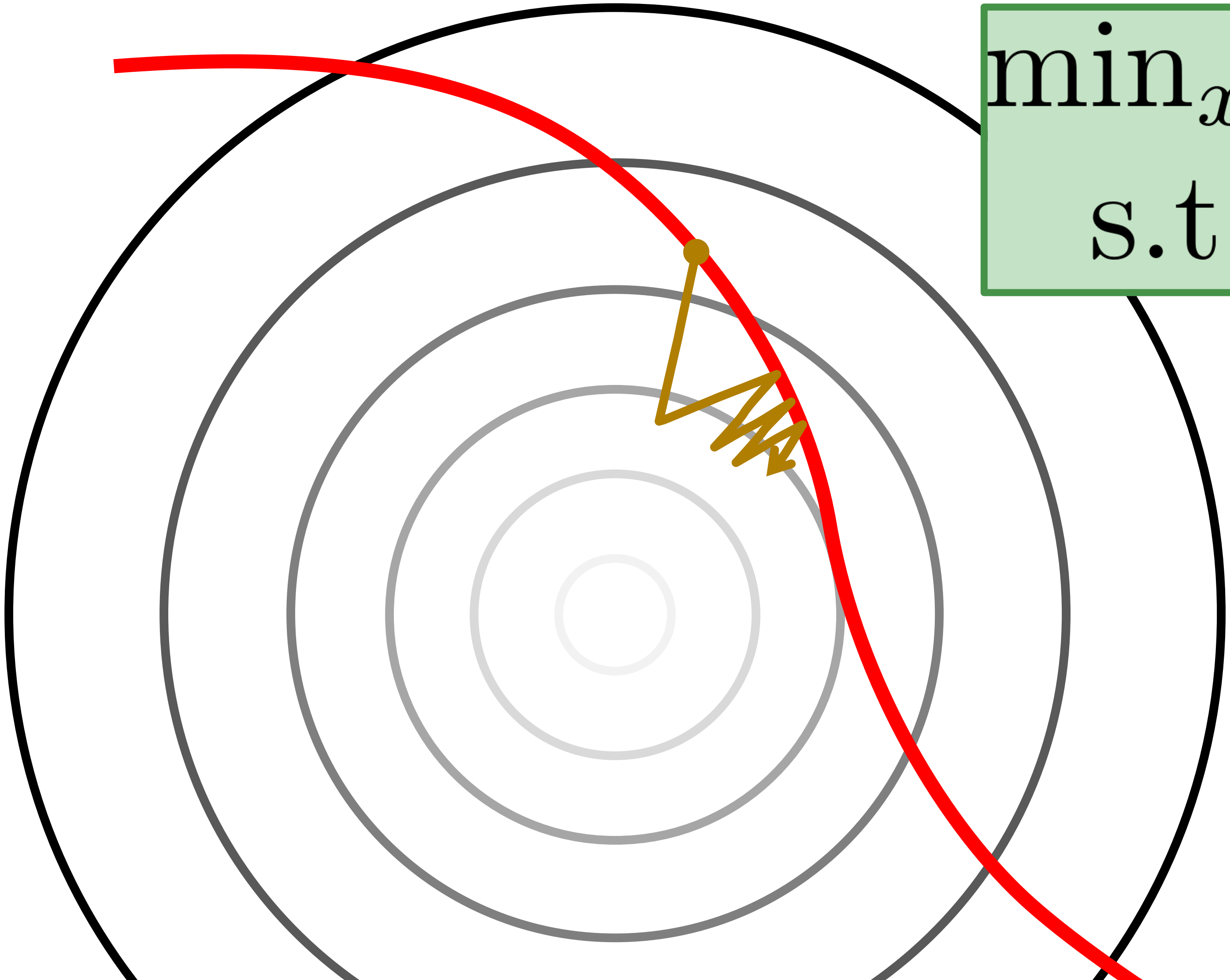
https://www.researchgate.net/figure/Visualization-of-the-spherical-and-Hopf-coordinates-on-SO3-using-angle-and-axis_fig1_45098059

Fundamental Disagreement



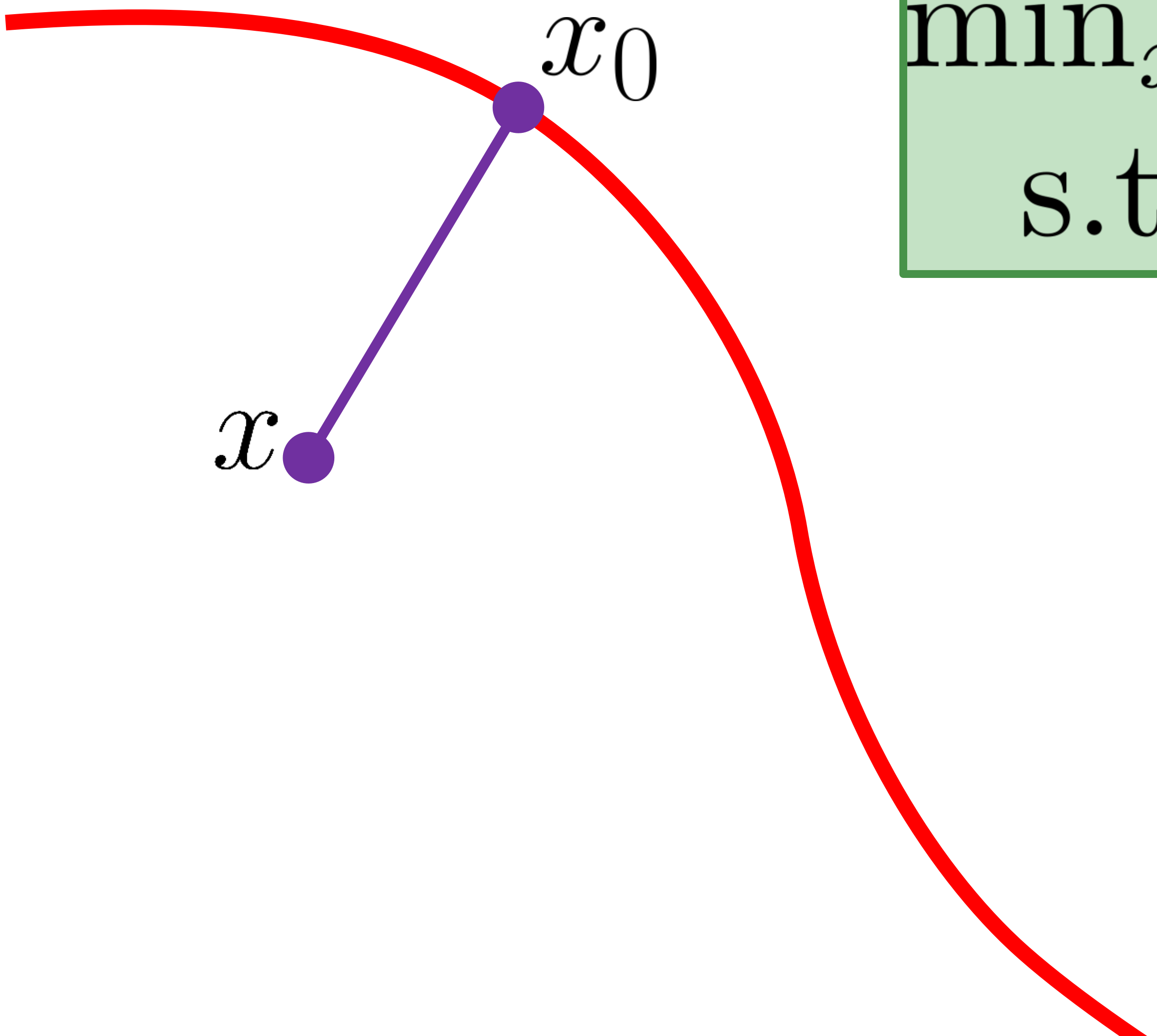
$$\min_x \quad f(x)$$
$$\text{s.t.} \quad g(x) = 0$$

Optimization Path: Step & Project



$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$

Projection Isn't Obvious!



$$\begin{aligned} \min_x \quad & \|x - x_0\|_2 \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$

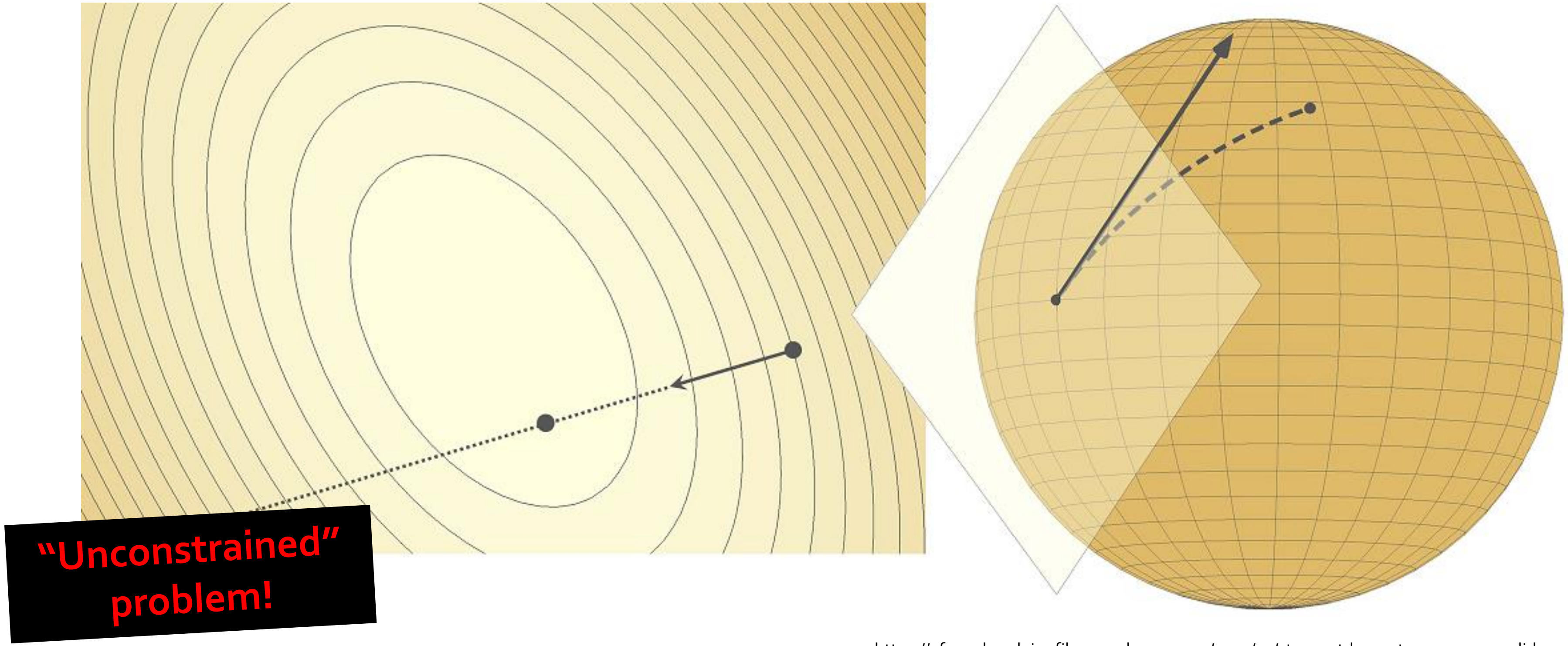
Intrinsic Perspective

http://opentranscripts.org/transcript/katherine-cross-at-the-conference-2015/

Optimization as a Lady Bug

https://www.shutterstock.com/video/clip-27741358-beautiful-tiny-ladybug-on-corn-leaf-slow-mo

Intrinsic Approach to Optimization



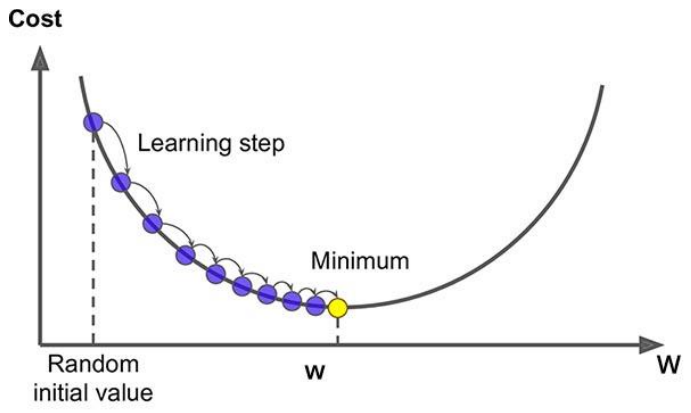
https://afonsobandeira.files.wordpress.com/2015/03/steepestdescent_compare_euclidean_sphere.png

Optimize without stepping off of the manifold

Starting Point

Gradient descent on $\mathbb{R}^n$

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$



https://saugatbhattarai.com.np/what-is-gradient-descent-in-machine-learning/

Q:

What are the **constituent parts** of gradient descent?

Starting Point

Gradient descent on $\mathbb{R}^n$

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

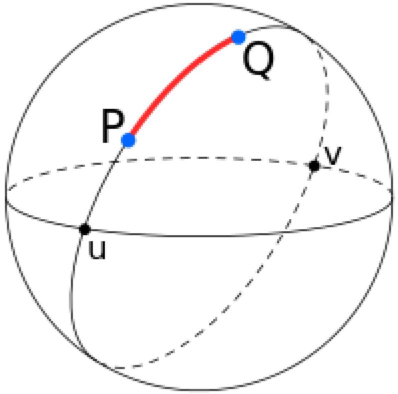
Gradient direction

Means of walking along the domain

First-Order Manifold Optimization

Gradient descent on $\mathbb{R}^n$

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$



Gradient direction

Means of walking along the domain

$$x_{k+1} = \exp_{x_k}(-\alpha_k \nabla f(x_k))$$

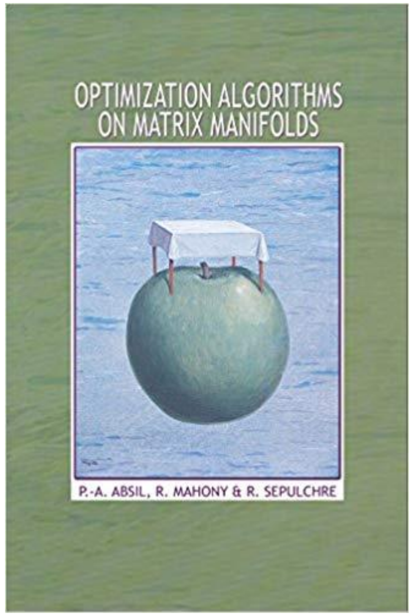
Manifold gradient descent (roughly)

Why Manifold Optimization?

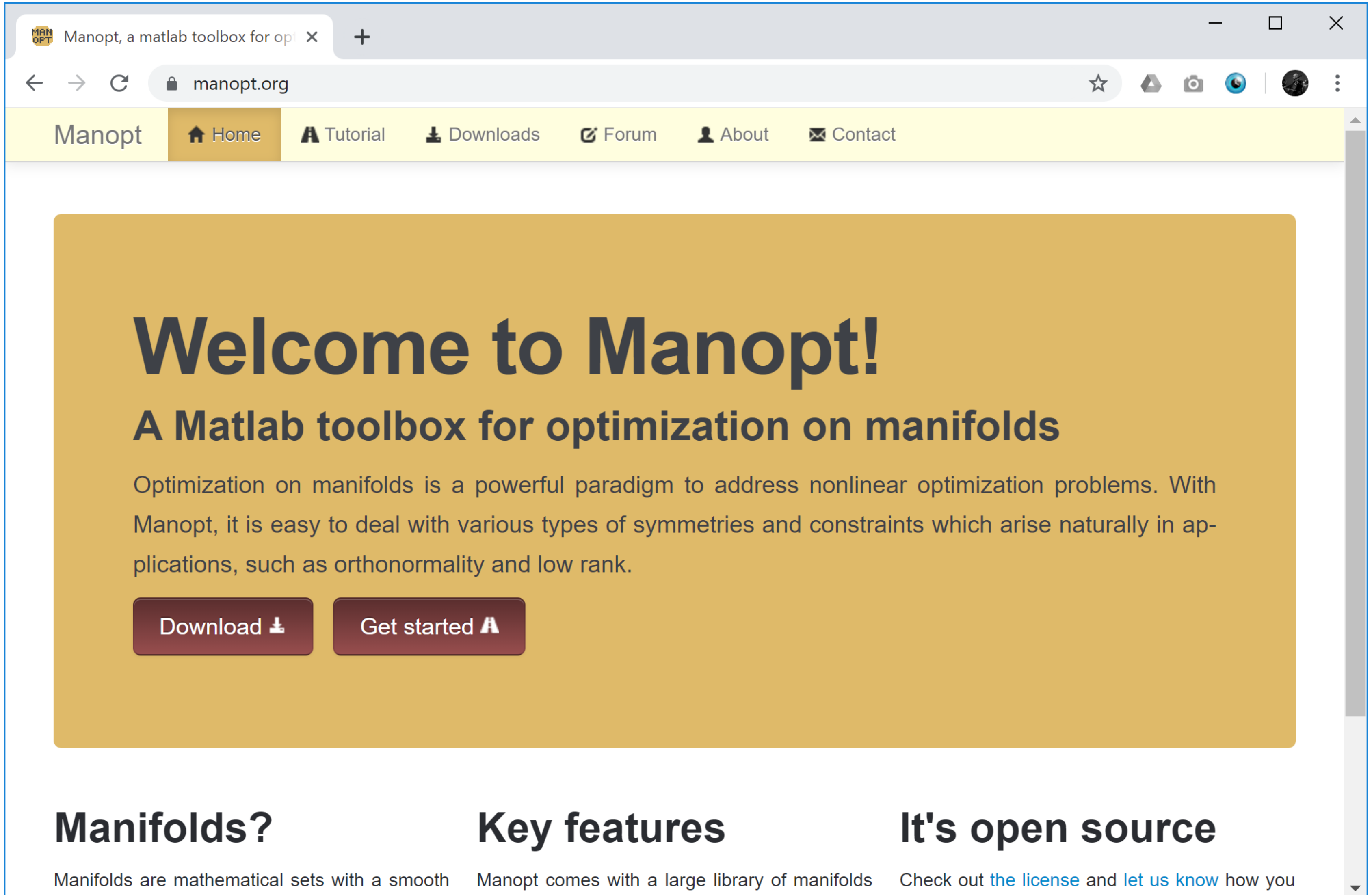
- Practical perspective:
  Better algorithms
  Automatic constraint satisfaction, specialized to the space

- Theoretical perspective:
  Elegant mathematical characterization
  Generalize convexity, gradient descent, …

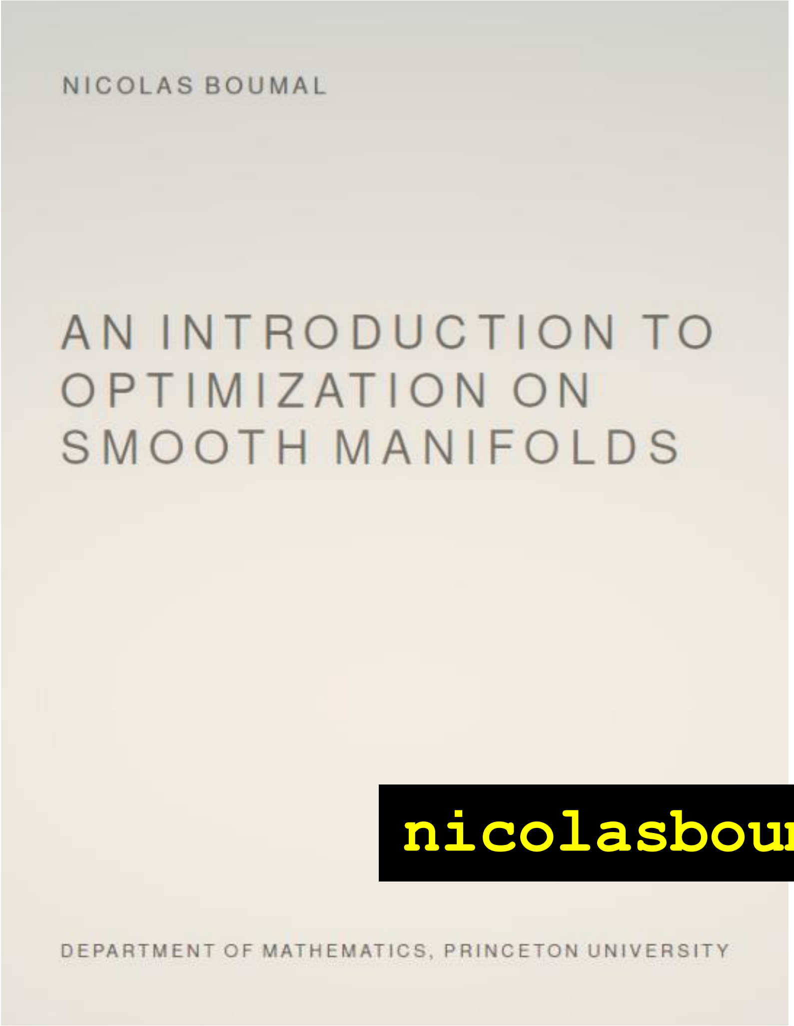
Comprehensive Introduction



Matlab Toolbox



Additional Resource



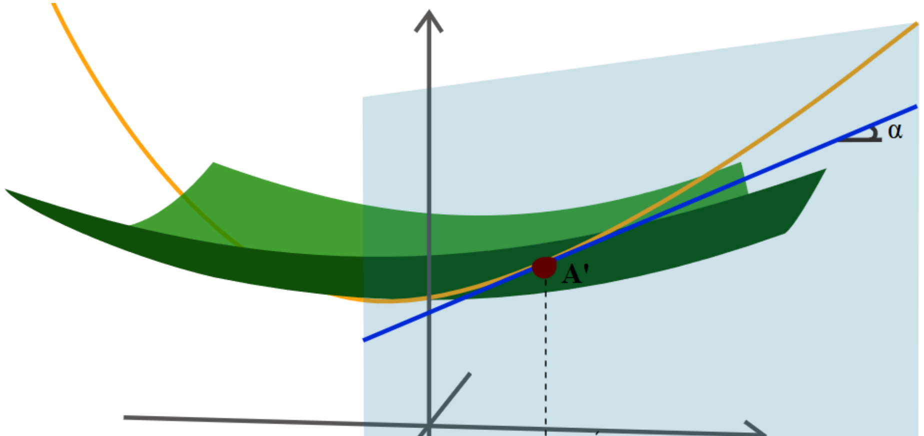
nicolasboumal.net/#book

Recall: Differential

$$df_{\mathbf{x}_0}(\mathbf{v}) := \lim_{h \to 0} \frac{f(\mathbf{x}_0 + h\mathbf{v}) - f(\mathbf{x}_0)}{h}$$

Proposition. df_{x_0} is a linear operator.

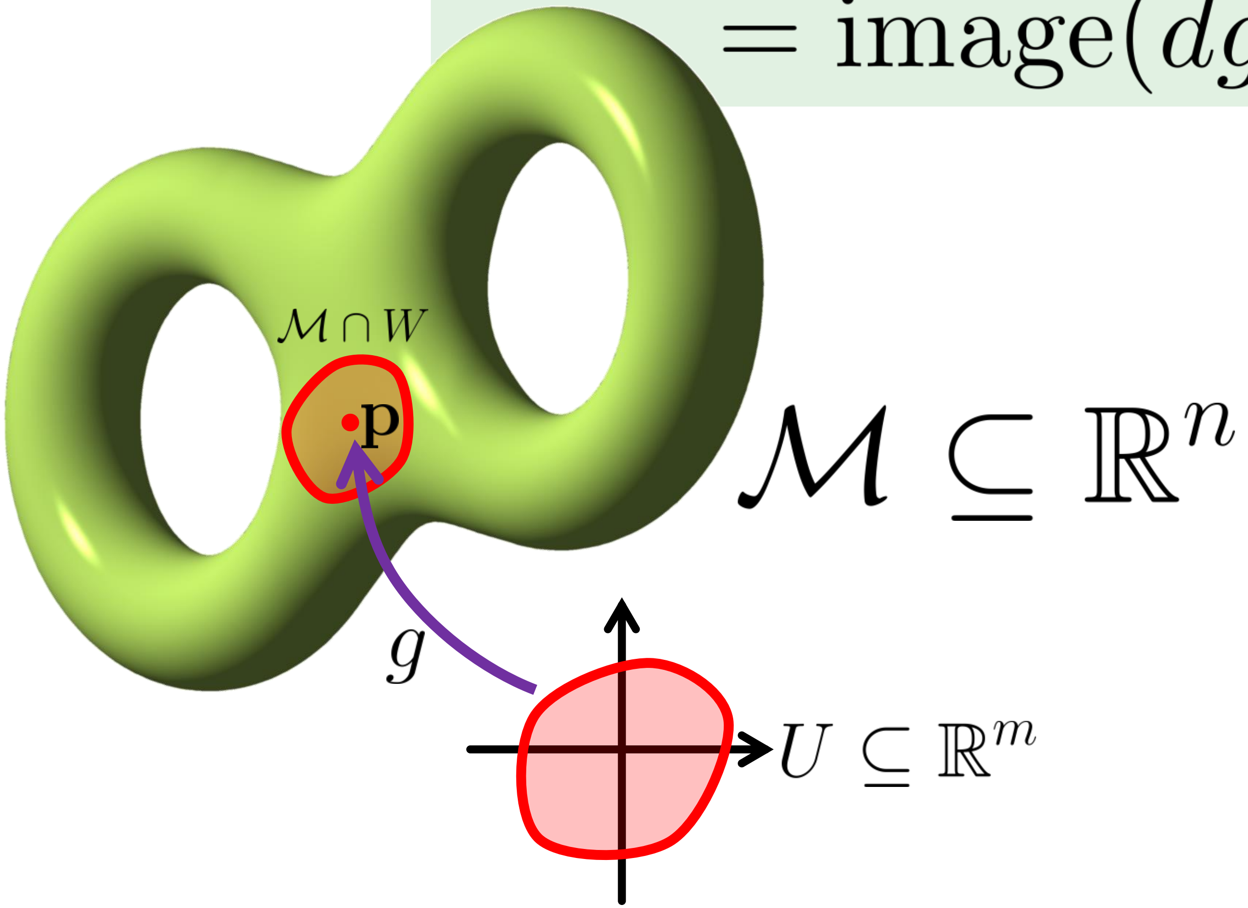
$$df_{\mathbf{x}_0}(\mathbf{v}) = Df(\mathbf{x}_0) \cdot \mathbf{v}$$



Note: Technically we derived the 1D version. Nothing changes!

Recall: Tangent Space

$$T_{\mathbf{p}}\mathcal{M} = \gamma'(0), \text{ where } \gamma(0) = \mathbf{p}$$
$$= \text{image}(dg_{g^{-1}(\mathbf{p})})$$

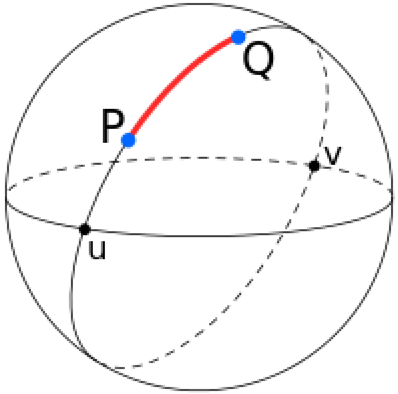


Skipping:
Independence of choice of g.

Back to Optimization

Gradient descent on $\mathbb{R}^n$

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$



Gradient direction

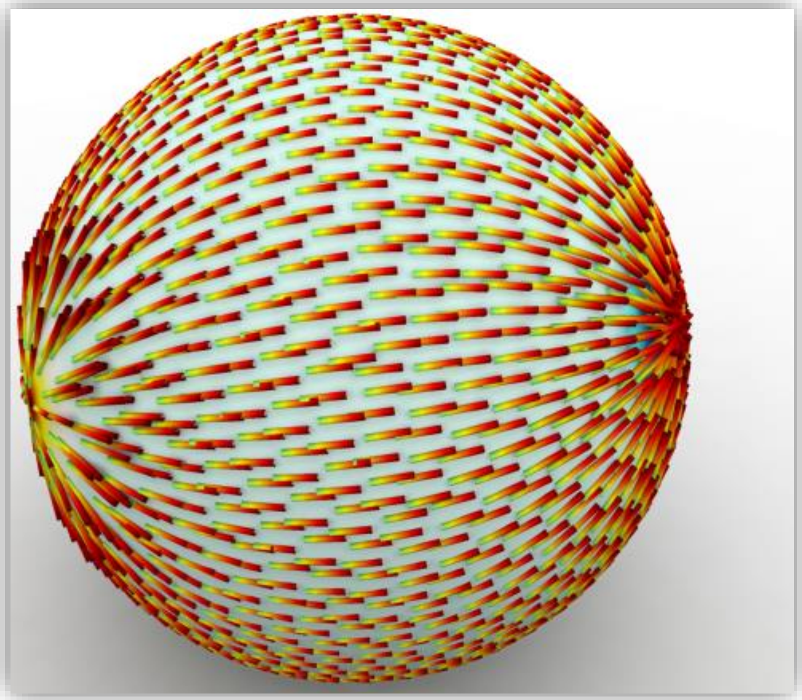
Means of walking along the domain

$$x_{k+1} = \exp_{x_k}(-\alpha_k \nabla f(x_k))$$

Manifold gradient descent (roughly)

Recall: Gradient

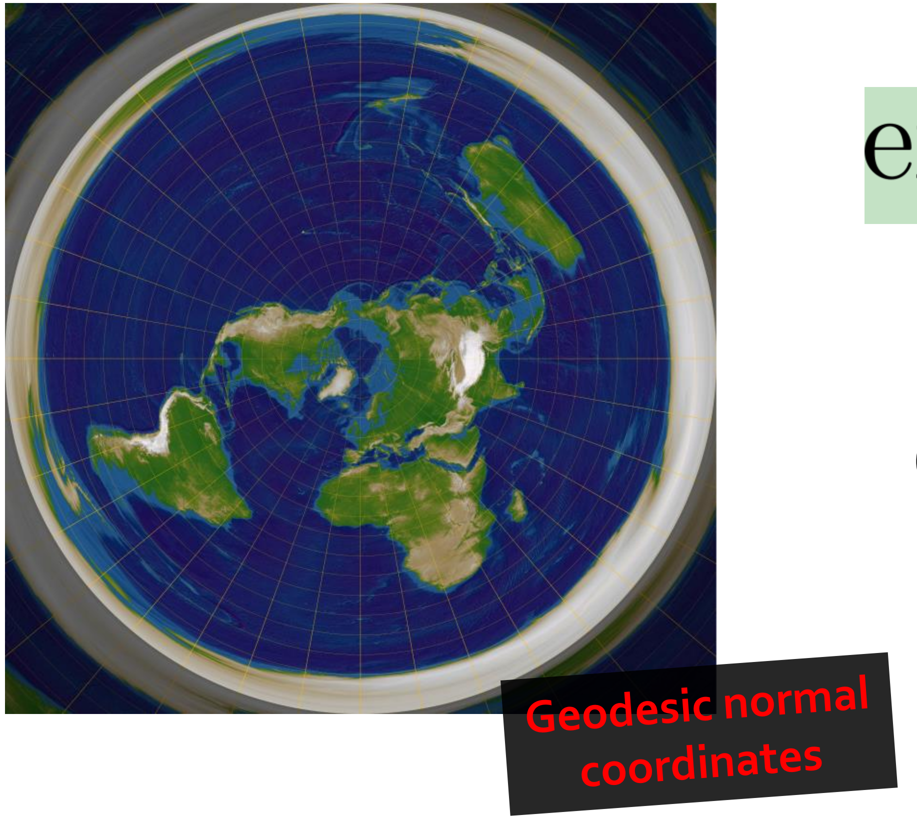
$$x_{k+1} = \exp_{x_k}(-\alpha_k \nabla f(x_k))$$



Proposition *For each* $\mathbf{p} \in \mathcal{M}$*, there exists a unique vector* $\nabla f(\mathbf{p}) \in T_{\mathbf{p}}\mathcal{M}$ *so that* $df_{\mathbf{p}}(\mathbf{v}) = \mathbf{v} \cdot \nabla f(\mathbf{p})$ *for all* $\mathbf{v} \in T_{\mathbf{p}}\mathcal{M}$*.*

Walking along the Manifold: Exponential Map

$$x_{k+1} = \exp_{x_k}(-\alpha_k \nabla f(x_k))$$

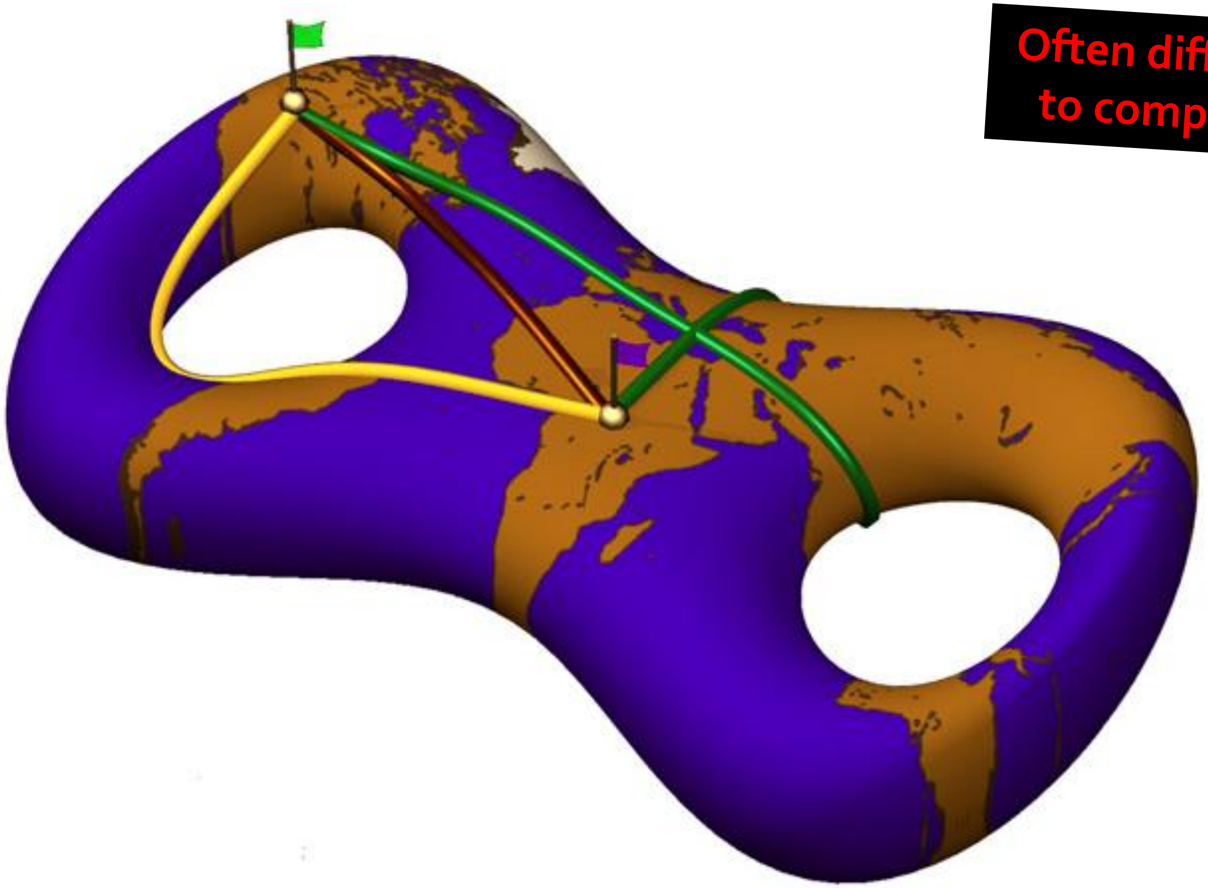


$$\exp_{\mathbf{p}}(\mathbf{v}) := \gamma_{\mathbf{v}}(1)$$

$\gamma_v(1)$ where γ_v is (unique) geodesic from *p* with velocity *v*.

https://en.wikipedia.org/wiki/Exponential_map_(Riemannian_geometry)

Geodesics are Complicated!

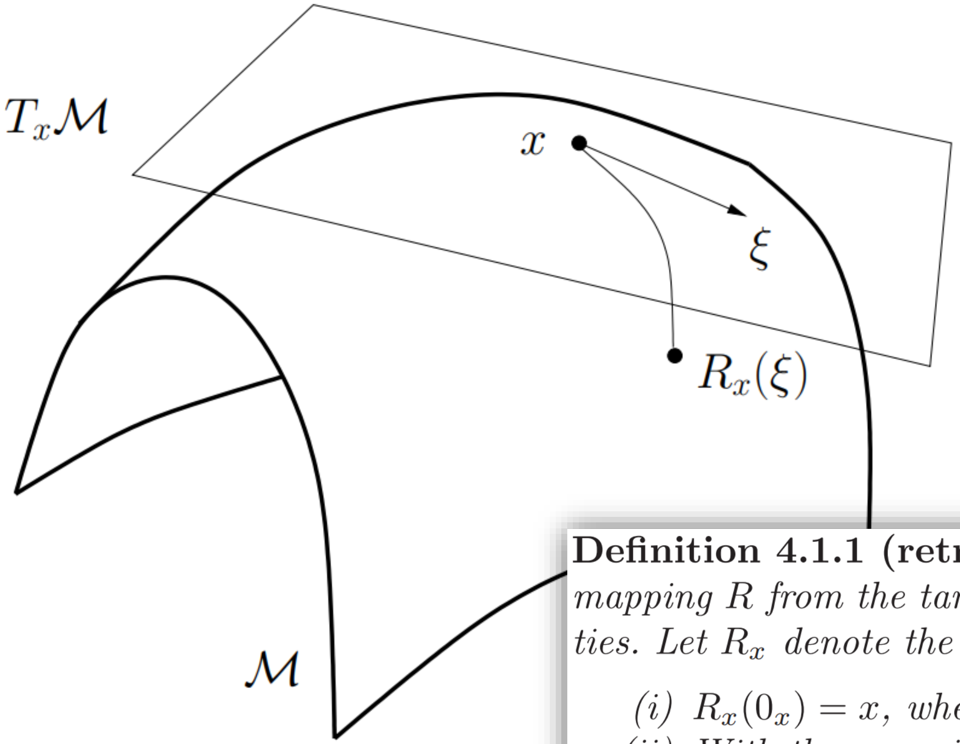


Often difficult to compute

Straightest Geodesics on Polyhedral Surfaces (Polthier and Schmies)

Weaker Notion: Retraction

$$x_{k+1} = \mathrm{retraction}_{x_k}(-\alpha_k \nabla f(x_k))$$



Definition 4.1.1 (retraction) *A* retraction *on a manifold $\mathcal{M}$ is a smooth mapping R from the tangent bundle $T\mathcal{M}$ onto $\mathcal{M}$ with the following properties. Let R_x denote the restriction of R to $T_x\mathcal{M}$.*

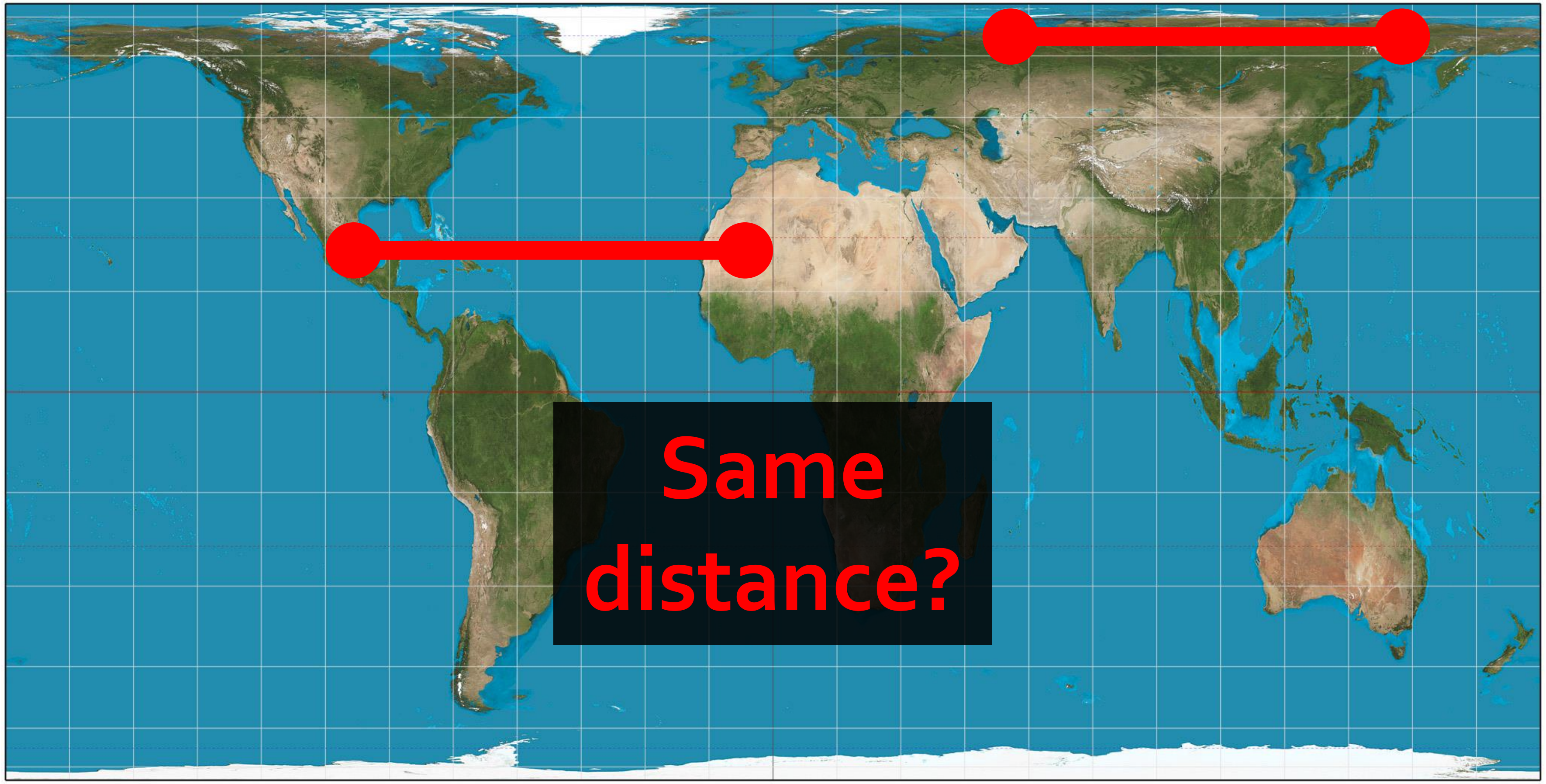
(i) $R_x(0_x) = x$, *where* 0_x *denotes the zero element of* $T_x\mathcal{M}$.
(ii) *With the canonical identification* $T_{0_x}T_x\mathcal{M} \simeq T_x\mathcal{M}$, R_x *satisfies*

$$\mathrm{D}R_x(0_x) = \mathrm{id}_{T_x\mathcal{M}},$$

where $\mathrm{id}_{T_x\mathcal{M}}$ *denotes the identity mapping on* $T_x\mathcal{M}$.

From "Optimization Algorithms on Matrix Manifolds" (Absil et al.)

More General Setting: Riemannian Manifold



Pair (M, g) of a differentiable manifold M and a pointwise positive definite inner product per point $g_p(\cdot,\cdot): T_pM \times T_pM \to \mathbb{R}$.

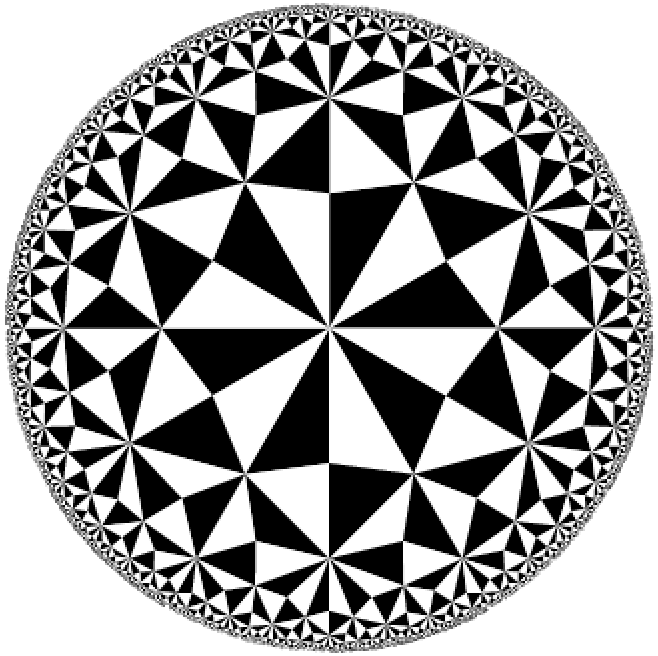
Riemannian Inner Product

$$g(\cdot,\cdot)_p : T_p\mathcal{M} \times T_p\mathcal{M} \rightarrow \mathbb{R}$$

Symmetric, bilinear, positive definite form

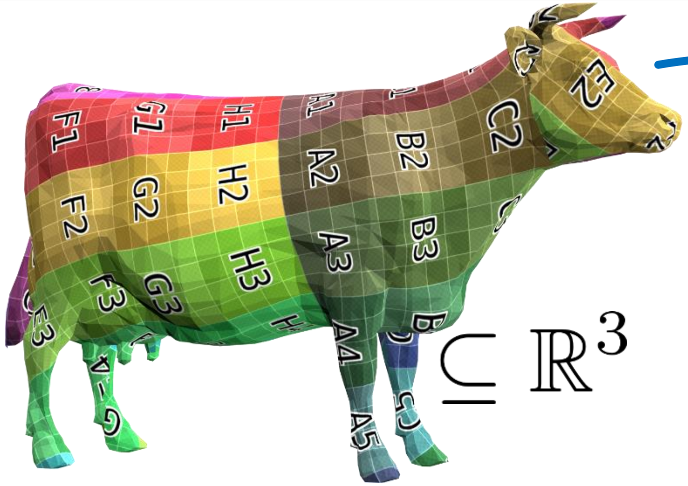
Example: Poincaré Disk

$$ds^2 = \frac{dx^2 + dy^2}{(1 - x^2 - y^2)^2}$$



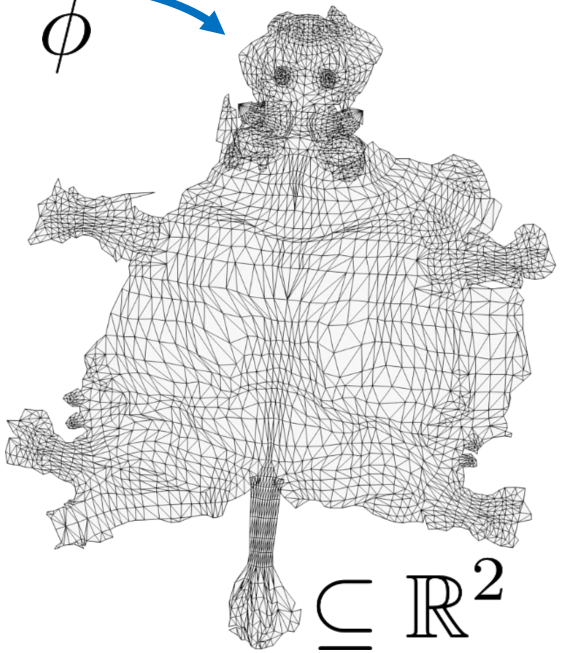
$$\cong$$

Example: Space of Parameterizations



$\subseteq \mathbb{R}^3$

ϕ



$\subseteq \mathbb{R}^2$

discretization

$$\begin{pmatrix} p_{1x} \\ p_{1y} \\ p_{2x} \\ \vdots \\ p_{nx} \\ p_{ny} \end{pmatrix} \in \mathbb{R}^{2|V|}$$

Claici, Bessmeltsev, Schaefer, and Solomon. "Isometry-Aware Preconditioning for Mesh Parameterization." SGP 2017, London.

Riemannian Gradient

- **Metric tensor $\boldsymbol{g} \in \mathbb{R}^{n \times n}$**
- **Gradient in coordinates $\boldsymbol{\nabla f} \in \mathbb{R}^{n}$**

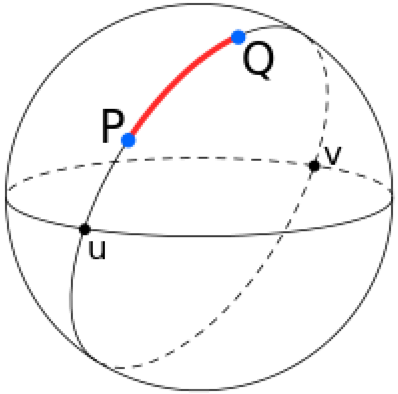
$$\nabla_g f = g^{-1} \nabla f$$

$$\nabla_g f = g^{-1} \nabla f$$

Riemannian Gradient Descent (the same!)

Gradient descent on $\mathbb{R}^n$

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$



Gradient direction

Means of walking along the domain

$$x_{k+1} = \exp_{x_k}(-\alpha_k \nabla f(x_k))$$

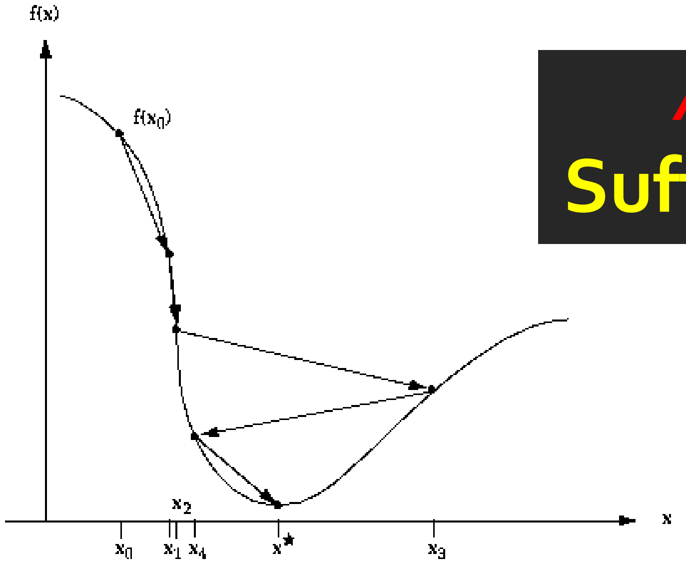
Manifold gradient descent (roughly)

Extension: Line Search

$$x_{k+1} = \text{retraction}_{x_k}(-\alpha_k \nabla f(x_k))$$

Identical strategies to Euclidean case:

- $\alpha_k = \frac{1}{k}$
- Backtracking
- 1D optimization



Advanced topic:
Sufficient decrease

https://www.phy.ornl.gov/csep/mo/node9.html

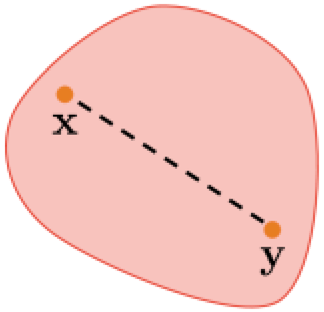
Extension: Newton's Method?

$$x_{k+1} \stackrel{?}{=} x_k - Hf(x_k)^{-1} \nabla f(x_k)$$

Tough to define
On manifolds gives a different search direction

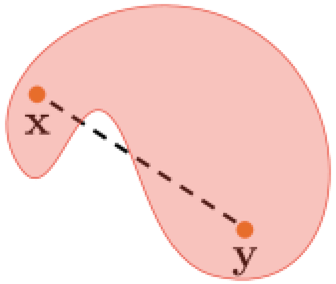
<omit>

Extension: Geodesic Convexity



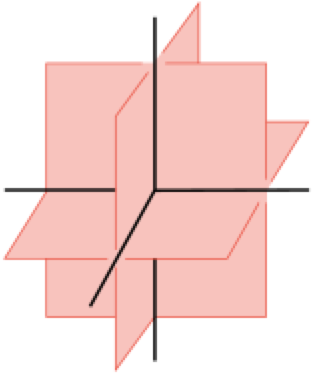
$\mathcal{C} \subseteq \mathbb{R}^d$

CONVEX SET



$\mathcal{C} \subseteq \mathbb{R}^d$

NON-CONVEX SET



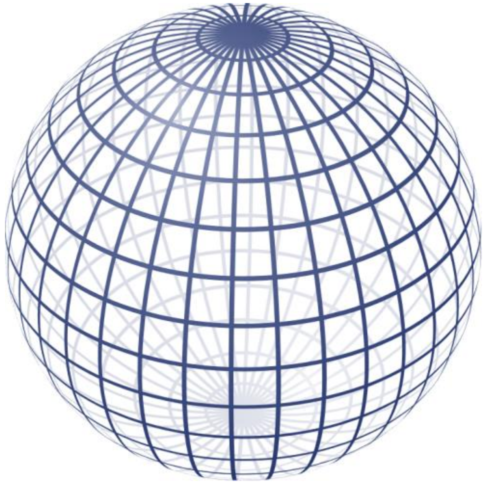
$\mathcal{B}_0(2) \subseteq \mathbb{R}^3$

NON-CONVEX SET

<omit>

https://www.groundai.com/project/non-convex-optimization-for-machine-learning/1

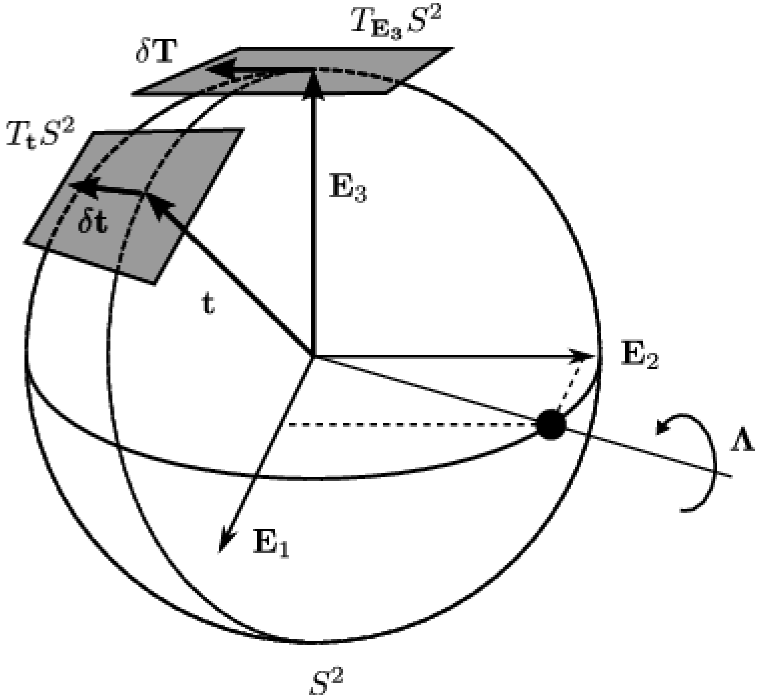
Example: Unit Sphere



$$S^{n-1} := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = 1\}$$

Tangent Space of Sphere

$$T_{\mathbf{p}}S^{n-1} = \{\mathbf{v} \in \mathbb{R}^n : \mathbf{v} \cdot \mathbf{p} = 0\}$$



https://math.stackexchange.com/questions/2219831/difference-of-vectors-living-in-different-tangent-spaces

Gradient

$$\text{Restriction of } f : \mathbb{R}^n \to \mathbb{R}$$

$$\nabla_{S^{n-1}} f(\mathbf{p}) = (I_{n\times n} - \mathbf{p}\mathbf{p}^\top)\nabla_{\mathbb{R}^n} f(\mathbf{p})$$

Project ambient gradient into the tangent plane

Retraction on Sphere: Two (Typical) Options

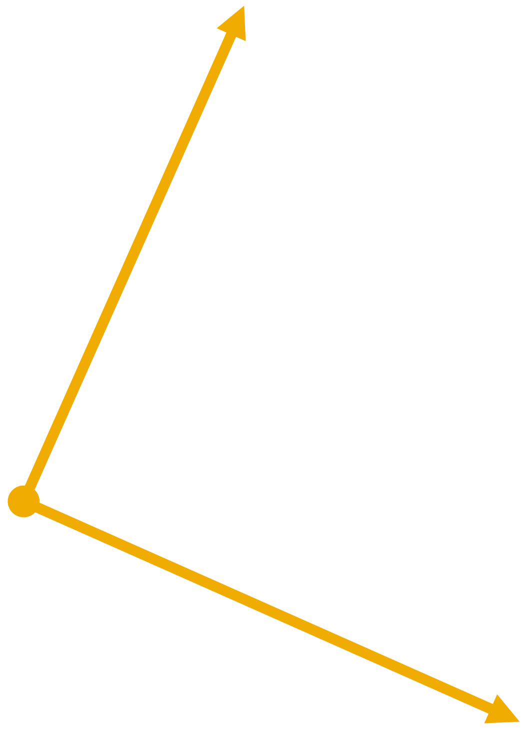
- Exponential map

$$\exp_{\mathbf{p}}(\mathbf{v}) = \mathbf{p}\cos\|\mathbf{v}\|_2 + \frac{\mathbf{v}\sin\|\mathbf{v}\|_2}{\|\mathbf{v}\|_2}$$

- Projection

$$R_{\mathbf{p}}(\mathbf{v}) = \frac{\mathbf{p}+\mathbf{v}}{\|\mathbf{p}+\mathbf{v}\|_2}$$

Example: Stiefel Manifold



$$V_k(\mathbb{R}^n) := \{X \in \mathbb{R}^{n \times k} : X^\top X = I_{k \times k}\}$$

$$V_k(\mathbb{R}^n) := \{X \in \mathbb{R}^{n\times k} : X^\top X = I_{k\times k}\}$$

Tangent Space and Retraction

$$T_X V_k(\mathbb{R}^n) = \{\xi \in \mathbb{R}^{n\times k} : \xi^\top X + X^\top \xi = 0_{k\times k}\}$$

$$R_X(\xi) := (X + \xi)(I_{k\times k} + \xi^\top \xi)^{-1/2}$$

Optimization Example: Rayleigh Quotient Minimization

$$\min_{\mathbf{x} \in S^{n-1}} \frac{1}{2} \mathbf{x}^\top A \mathbf{x}$$

Assume A is symmetric

On the board:

- Relationship to eigenproblems
- Intrinsic gradient
- First-order algorithm
- Extension: refining eigenvector estimates

Recall: Two (Typical) Options

- **Exponential map**

$$\exp_{\mathbf{p}}(\mathbf{v}) = \mathbf{p}\cos\|\mathbf{v}\|_2 + \frac{\mathbf{v}\sin\|\mathbf{v}\|_2}{\|\mathbf{v}\|_2}$$

- **Projection**

$$R_{\mathbf{p}}(\mathbf{v}) = \frac{\mathbf{p}+\mathbf{v}}{\|\mathbf{p}+\mathbf{v}\|_2}$$

$$\min_{\mathbf{x} \in S^{n-1}} \frac{1}{2} \mathbf{x}^\top A \mathbf{x}$$

Optimization Example: Regularized PCA

$$\max_{X \in V_k(\mathbb{R}^n)} \|X^\top A\|_{\text{Fro}}^2$$

On the board:

- Relationship to PCA
- Extensions: Robust PCA, regularized PCA
- Intrinsic gradient
- First-order algorithm

$$\max_{X \in V_k(\mathbb{R}^n)} \|X^\top A\|_{\mathrm{Fro}}^2$$

What's Next: Many (!) Variations of PCA

Method	Objective $f_X(M)$	Manifold $\mathcal{M}$	Mapping $Y = PX$
PCA (§3.1.1)	$\|\|X - MM^\top X\|\|_F^2$	$\mathcal{O}^{d\times r}$	$M^\top X$
MDS (§3.1.2)	$\sum_{i,j} \left(d_X(x_i, x_j) - d_Y(M^\top x_i, M^\top x_j)\right)^2$	$\mathcal{O}^{d\times r}$	$M^\top X$
LDA (§3.1.3)	$\frac{\mathrm{tr}(M^\top \Sigma_B M)}{\mathrm{tr}(M^\top \Sigma_W M)}$	$\mathcal{O}^{d\times r}$	$M^\top X$
Traditional CCA (§3.1.4)	$\mathrm{tr}\left(M_a^\top (X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2} M_b\right)$	$\mathcal{O}^{d_a\times r} \times \mathcal{O}^{d_b\times r}$	$M_a^\top \left(X_a X_a^\top\right)^{-1/2} X_a,$ $M_b^\top \left(X_b X_b^\top\right)^{-1/2} X_b$
Orthogonal CCA (§3.1.4)	$\frac{\mathrm{tr}\left(M_a^\top X_a X_b^\top M_b\right)}{\sqrt{\mathrm{tr}\left(M_a^\top X_a X_a^\top M_a\right)\mathrm{tr}\left(M_b^\top X_b X_b^\top M_b\right)}}$	$\mathcal{O}^{d_a\times r} \times \mathcal{O}^{d_b\times r}$	$M_a^\top X_a$, $M_b^\top X_b$
MAF (§3.1.5)	$\frac{\mathrm{tr}(M^\top \Sigma_\delta M)}{\mathrm{tr}(M^\top \Sigma M)}$	$\mathcal{O}^{d\times r}$	$M^\top X$
SFA (§3.1.6)	$\mathrm{tr}(M^\top \dot{X}\dot{X}^\top M)$	$\mathcal{O}^{d\times r}$	$M^\top X$
SDR (§3.1.7)	$\mathrm{tr}\left(\bar{K}_Z \left(\bar{K}_{M^\top X} + n\epsilon I\right)^{-1}\right)$	$\mathcal{O}^{d\times r}$	$M^\top X$
LPP (§3.1.8)	$\mathrm{tr}\left(M^\top (XDX^\top)^{-\top/2} XLX^\top (XDX^\top)^{-1/2} M\right)$	$\mathcal{O}^{d\times r}$	$M^\top (XDX^\top)^{-\top/2} X$
UICA (§3.2.1)	$\frac{1}{2}\log\|M^\top M\| + \frac{1}{n}\sum_{i=1}^n \sum_{k=1}^r \log f_\theta\left(m_k^\top x_n\right)$	$\mathbb{R}^{d\times r}$	$M^\top X$
PPCA (§3.2.2)	$\log\|MM^\top + \sigma^2 I\| + \mathrm{tr}\left(XX^\top (MM^\top + \sigma^2 I)^{-1}\right)$	$\mathbb{R}^{d\times r}$	$M^\top (MM^\top + \sigma^2 I)^{-1} X$
FA (§3.2.3)	$\log\|MM^\top + D\| + \mathrm{tr}\left(XX^\top (MM^\top + D)^{-1}\right)$	$\mathbb{R}^{d\times r}$	$M^\top (MM^\top + D)^{-1} X$
LR (§3.2.4)	$\|\|X_{out} - MX_{in}\|\|_F^2 + \lambda\|\|M\|\|_p$	$\mathbb{R}^{d\times r}$	$SV^\top X_{in}$ for $M = USV^\top$
DML (§3.2.5)	$\sum_{i,j\in\eta(i)} \left\{ d_M(x_i, x_j)^2 + \lambda \sum_\ell \mathbb{1}(z_i \neq z_\ell) \left[1 + d_M(x_i, x_j)^2 - d_M(x_i, x_\ell)^2\right]_+ \right\}$	$\mathbb{R}^{d\times r}$	$M^\top X$

From "Linear Dimensionality Reduction: Survey, Insights, and Generalizations"
(Cunningham & Ghahramani; JMLR 2015)

What's Next: Efficient Semidefinite Programming

Samuel Burer · Renato D.C. Monteiro

A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization

Received: March 22, 2001 / Accepted: August 30, 2002
Published online: December 9, 2002 – © Springer-Verlag 20

Abstract. In this paper, we present a nonlinear programmi
(SDPs) in standard form. The algorithm's distinguishing fe
symmetric, positive semidefinite variable X of the SDP with
ization $X = RR^T$. The rank of the factorization, i.e., the num
to enhance computational speed while maintaining equivalenc
the convergence of the algorithm are derived, and encouragin
problems are also presented.

Key words. semidefinite programming – low-rank factori
Lagrangian – limited memory BFGS

1. Introduction

In the past few years, the topic of semidefinite
considerable attention in the optimization co
included the investigation of theoretically efficien
tical implementation codes, and the exploration

Deterministic guarantees for Burer–Monteiro factorizations of smooth semidefinite programs

NICOLAS BOUMAL
Mathematics Department and Program in Applied and Computational Mathematics, Princeton University

VLADISLAV VORONINSKI
Helm.ai

AND

AFONSO S. BANDEIRA
Department of Mathematics and Center for Data Science, Courant Institute of Mathematical Sciences, New York University

Abstract

We consider semidefinite programs (SDPs) with equality constraints. The variable to be optimized is a positive semidefinite matrix X of size n. Following the Burer–Monteiro approach, we optimize a factor Y of size $n \times p$ instead, such that $X = YY^\top$. This ensures positive semidefiniteness at no cost and can reduce the dimension of the problem if p is small, but results in a non-convex optimiza-

What’s Next: Distance Completion & Embedding

Low-rank optimization for distance matrix completion

B. Mishra, G. Meyer and R. Sepulchre

Abstract— **This paper addresses the problem of low-rank distance matrix completion. This problem amounts to recover the missing entries of a distance matrix when the dimension of the data embedding space is possibly unknown but small compared to the number of considered data points. The focus is on high-dimensional problems. We recast the considered problem into an optimization problem over the set of low-rank positive semidefinite matrices and propose two efficient algorithms for low-rank distance matrix completion. In addition, we propose a strategy to determine the dimension of the embedding space. The resulting algorithms scale to high-dimensional problems and monotonically converge to a global solution of the problem. Finally, numerical experiments illustrate the good performance of the proposed algorithms on benchmarks.**

This is the pre-print version of [1].

I. INTRODUCTION

Completing the missing entries of a matrix under low-rank constraint is a fundamental and recurrent problem in many modern engineering applications (see [2] and references therein). Recently, the problem has gained much popularity thanks to collaborative filtering applications and the Netflix challenge [3]

a restrictive set of given distances. Inference on the unknown entries is possible thanks to the low-rank property which models the redundancy between the available data.

A closely related problem is multidimensional scaling (MDS) for which all pairwise distances are available up front. A solution to this problem is the classical multidimensional scaling algorithm (CMDS), which relies on singular value decomposition to find a globally optimum embedding of fixed-rank. The CMDS algorithm minimizes the total quadratic error on scalar products between data points. Other algorithms have focused on variant cost functions, see the paper [10] for a survey in this area.

In contrast to the classical multidimensional scaling formulation, the problem of Euclidean distance matrix completion involves missing distances. The problem can be considered as a variant of multidimensional scaling problem with binary weights [10], [11]. The low-rank distance matrix completion problem is known to be NP-hard in general [12], [13], but convex relaxations have been proposed to render the problem tractable [14], [15]. Typical convex relaxations cast the EDM completion problem into a convex optimization

What's Next: Low-Rank Completion

SIAM J. OPTIM.
Vol. 23, No. 2, pp. 1214–1236

LOW-RANK MATRIX COMPLETION BY RIEMANNIAN OPTIMIZATION*

BART VANDEREYCKEN†

Abstract. The matrix completion problem consists of finding or approximating a low-rank matrix based on a few samples of this matrix. We propose a new algorithm for matrix completion that minimizes the least-square distance on the sampling set over the Riemannian manifold of fixed-rank matrices. The algorithm is an adaptation of classical nonlinear conjugate gradients, developed within the framework of retraction-based optimization on manifolds. We describe all the necessary objects from differential geometry necessary to perform optimization over this low-rank matrix manifold, seen as a submanifold embedded in the space of matrices. In particular, we describe how metric projection can be used as retraction and how vector transport lets us obtain the conjugate search directions. Finally, we prove convergence of a regularized version of our algorithm under the assumption that the restricted isometry property holds for incoherent matrices throughout the iterations. The numerical experiments indicate that our approach scales very well for large-scale problems and compares favorably with the state-of-the-art, while outperforming most existing solvers.

Key words. matrix completion, low-rank matrices, optimization on manifolds, differential geometry, nonlinear conjugate gradients, Riemannian manifolds, Newton

AMS subject classifications. 15A83, 65K05, 53B21

DOI. 10.1137/110845768

1. Introduction. Let $A \in \mathbb{R}^{m \times n}$ be an $m \times n$ matrix that is only known on a subset Ω of the complete set of entries $\{1, \ldots, m\} \times \{1, \ldots, n\}$. The low-rank matrix completion problem [16] consists of finding the matrix with lowest rank that agrees with A on Ω:

What's Next: Synchronization

Article

The International Journal of Robotics Research
2019, Vol. 38(2-3) 95–125

DOI: 10.1177/0278364918784361
journals.sagepub.com/home/ijr
SAGE

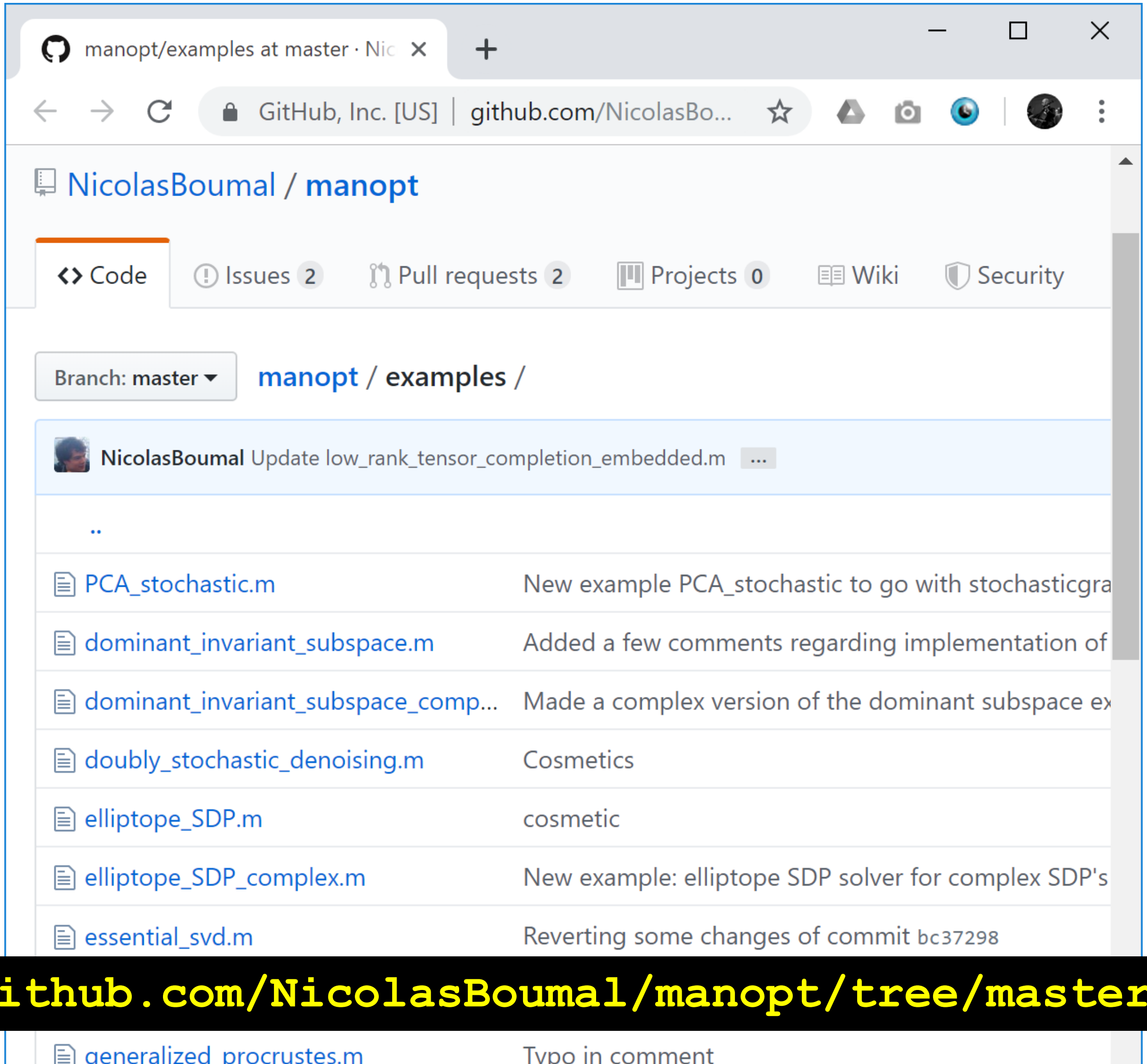
SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group

David M Rosen[1], Luca Carlone[2], Afonso S Bandeira[3], and John J Leonard[4]

Abstract

Many important geometric estimation problems naturally take the form of synchronization over the special Euclidean group*: estimate the values of a set of unknown group elements* $x_1, \ldots, x_n \in \mathrm{SE}(d)$ *given noisy measurements of a subset of their pairwise relative transforms* $x_i^{-1}x_j$. *Examples of this class include the foundational problems of pose-graph simultaneous localization and mapping (SLAM) (in robotics), camera motion estimation (in computer vision), and sensor network localization (in distributed sensing), among others. This inference problem is typically formulated as a non-convex maximum-likelihood estimation that is computationally hard to solve in general. Nevertheless, in this paper we present an algorithm that is able to efficiently recover* certifiably globally optimal *solutions of the special Euclidean synchronization problem in a non-adversarial noise regime. The crux of our approach is the development of a semidefinite relaxation of the maximum-likelihood estimation (MLE) whose minimizer provides an* exact *maximum-likelihood estimate so long as the magnitude of the noise corrupting the available measurements falls below a certain critical threshold; furthermore, whenever exactness obtains, it is possible to* verify *this fact* a posteriori, *thereby* certifying *the optimality of the recovered estimate. We develop a specialized optimization scheme for solving large-scale instances of this semidefinite relaxation by exploiting its low-rank, geometric, and graph-theoretic structure to reduce it to an equivalent optimization problem defined on a low-dimensional Riemannian manifold, and then design a Riemannian truncated-Newton trust-region method to solve*

Many Examples



https://github.com/NicolasBoumal/manopt/tree/master/examples

Optimization on Manifolds

Justin Solomon

6.838: Shape Analysis

Spring 2021

Extra: Manifold Methods for PCA

Justin Solomon

6.838: Shape Analysis

Spring 2021