

Optimization on Manifolds

Justin Solomon
MLSS 2019



Common Constraints in ML

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

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

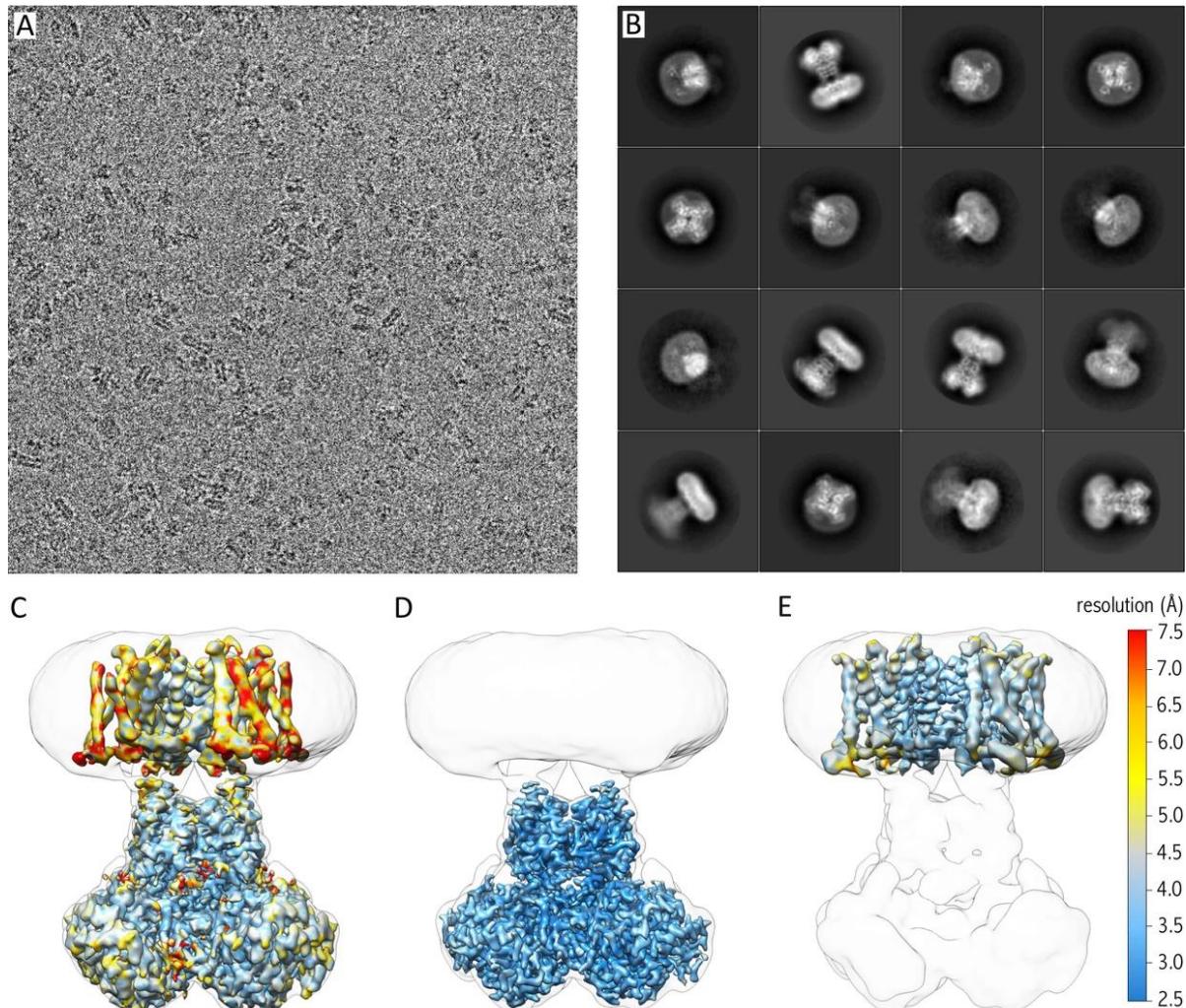
Example: Structure-from-Motion



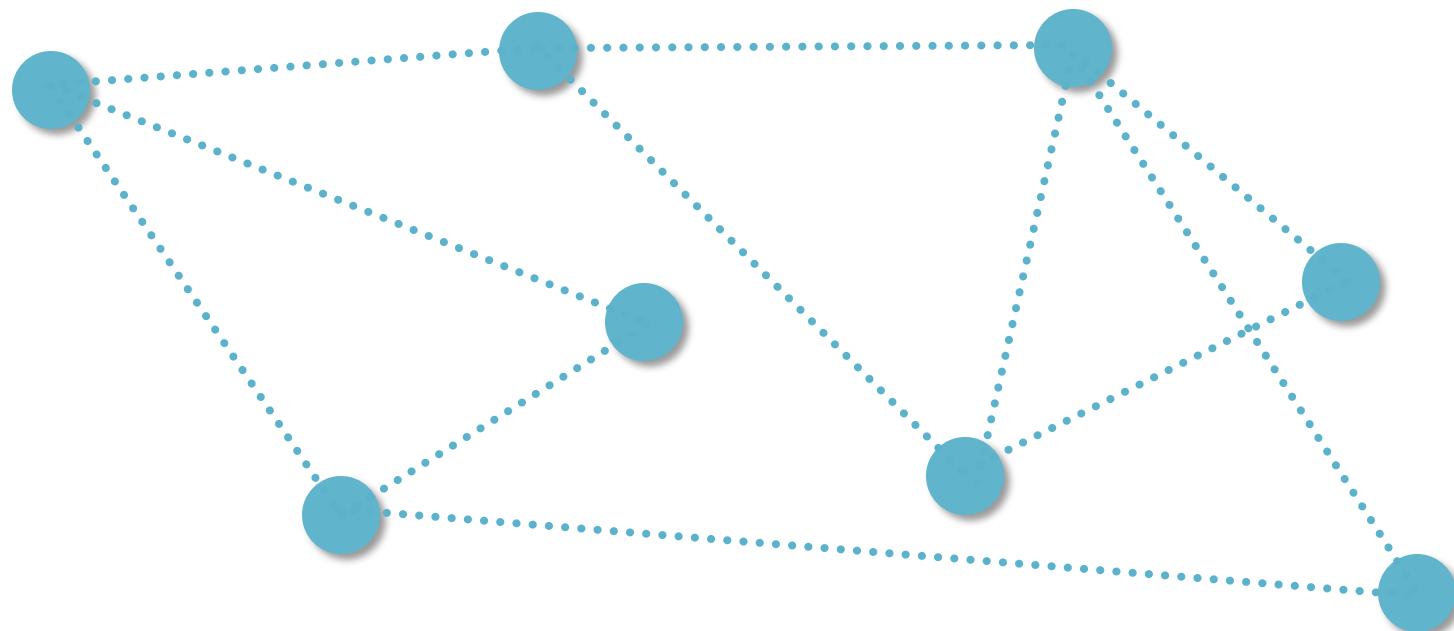
Search space: set of rotations

SfM (Princeton Vision & Robotics group)

Cryo-EM



Example: Sensor Network Localization



Search space: cloud of points, up to rigid motion

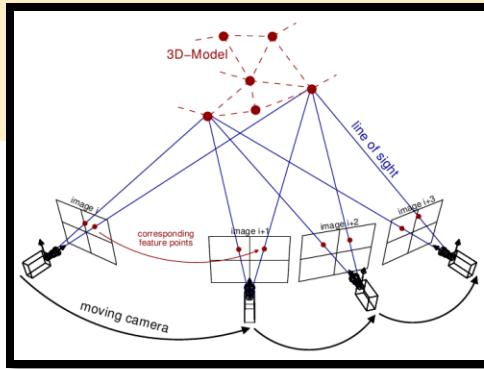
Typical Approach

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

↑
constraint

Example:

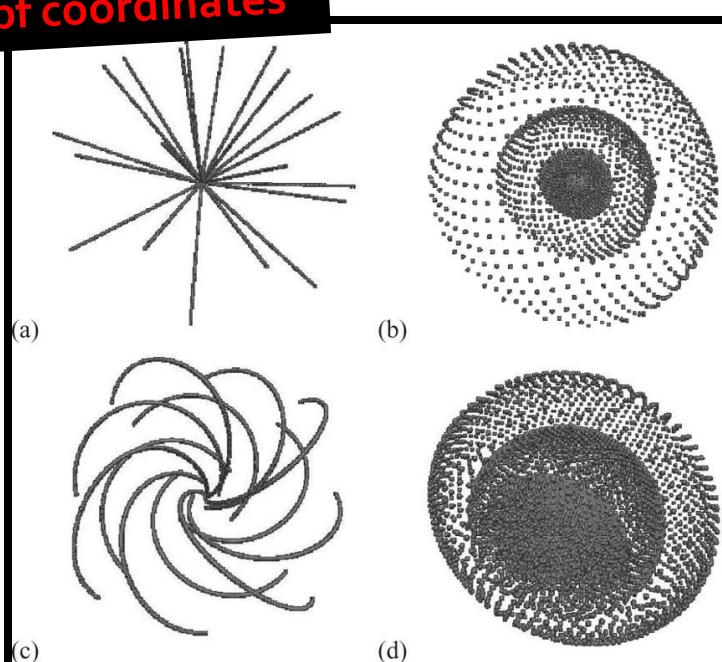
$$\min_{R \in \mathrm{SO}(3)} f(R) \longmapsto \begin{array}{l} \min_{R^\top R = I_3} f(R) \\ \det(R) = 1 \end{array}$$



Challenges

$$\begin{aligned} & \min f(R) \\ & R^\top R = I_3 \\ & \det(R) = 1 \end{aligned}$$

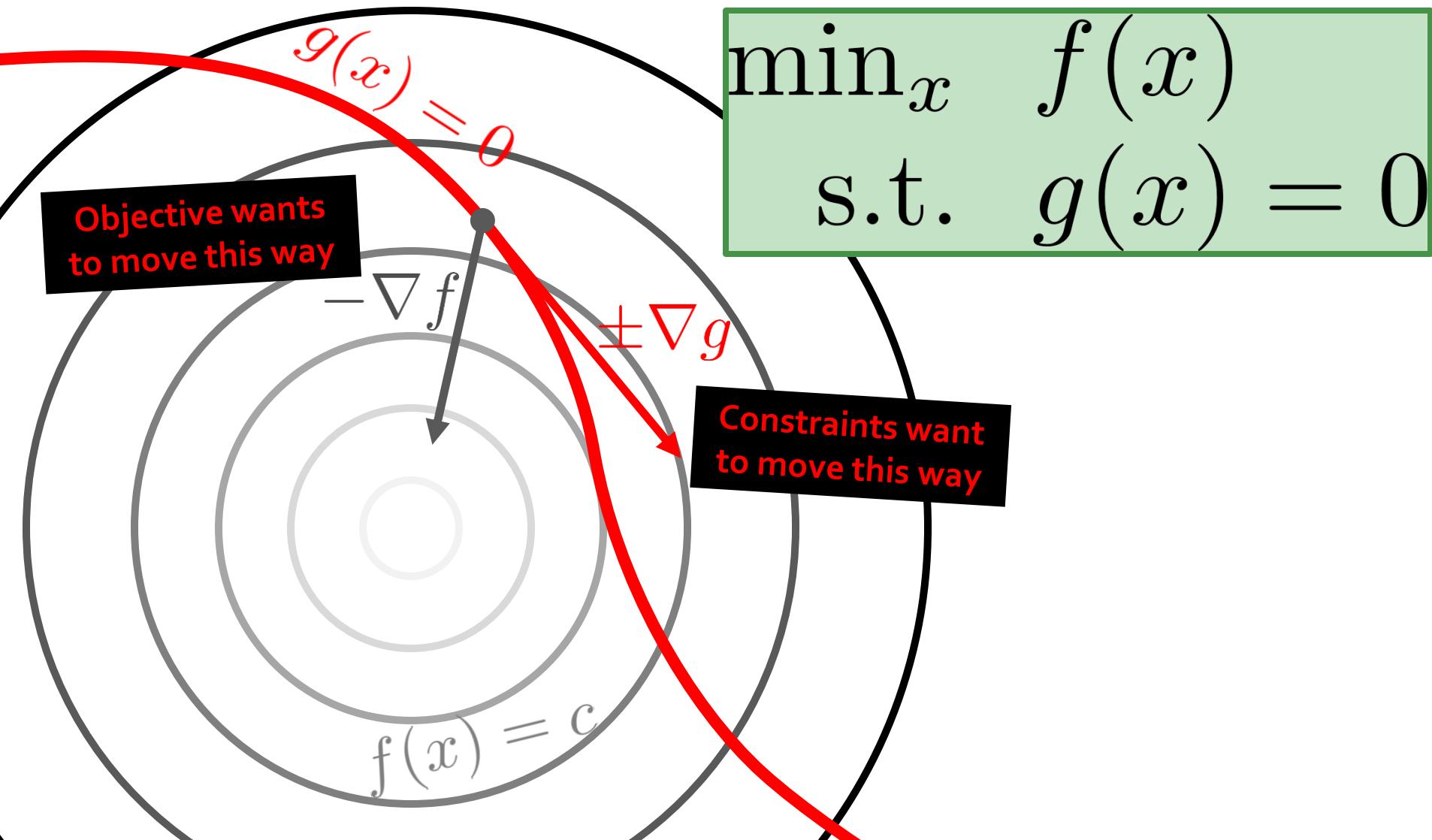
"Hopf coordinates"



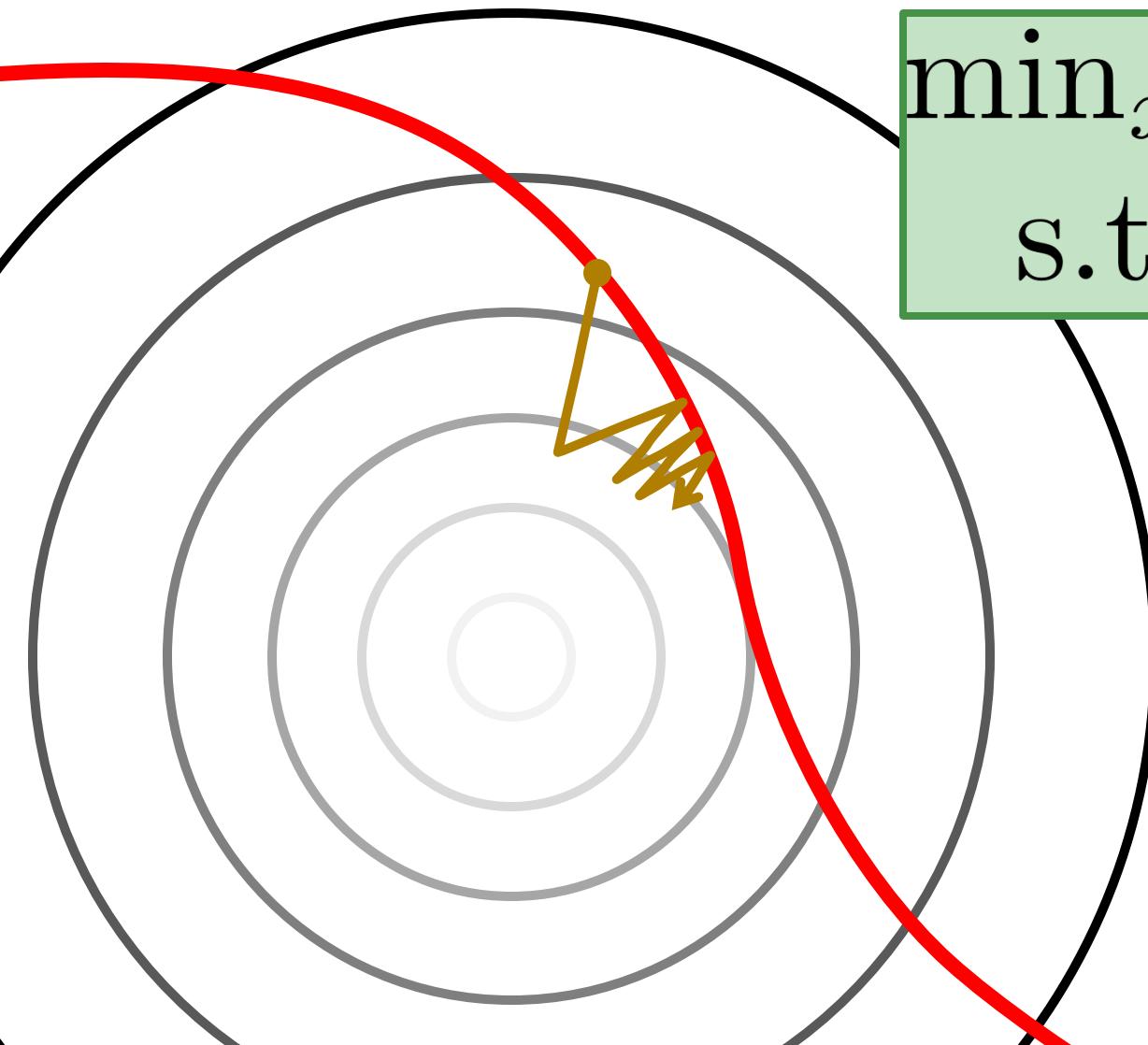
Constraints cut out a
nonconvex, curved set in $\mathbb{R}^{3 \times 3}$

Naively: Degree 2 & 3 polynomials

Fundamental Disagreement

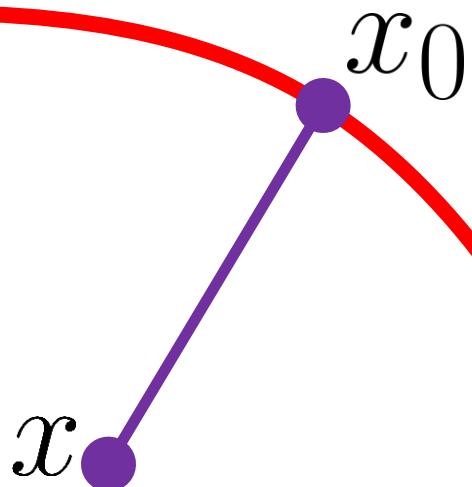


Optimization Path: Step & Project



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

Projection Isn't Obvious!



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

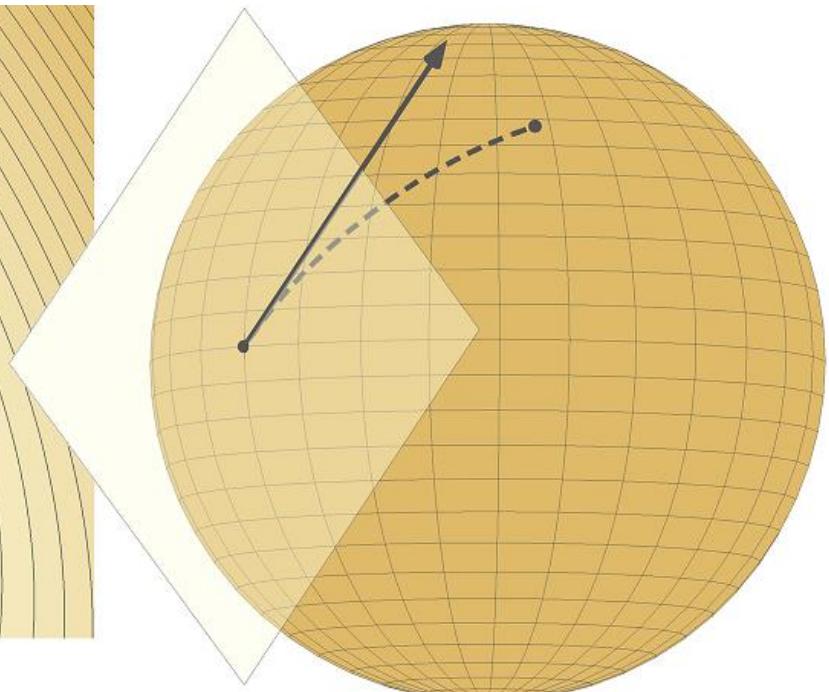
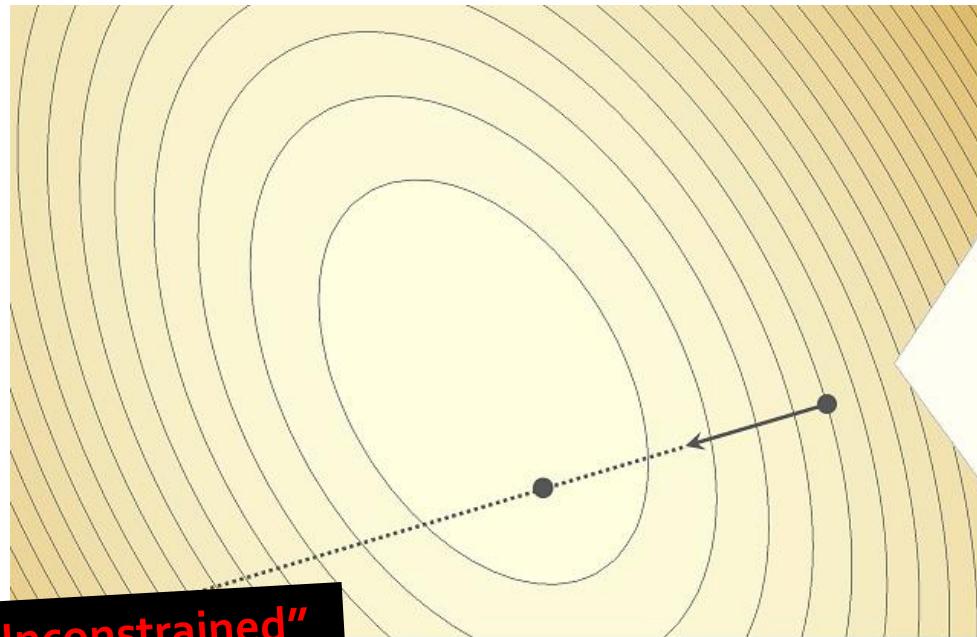
Intrinsic Perspective



Optimization as a Lady Bug



Intrinsic Approach to Optimization



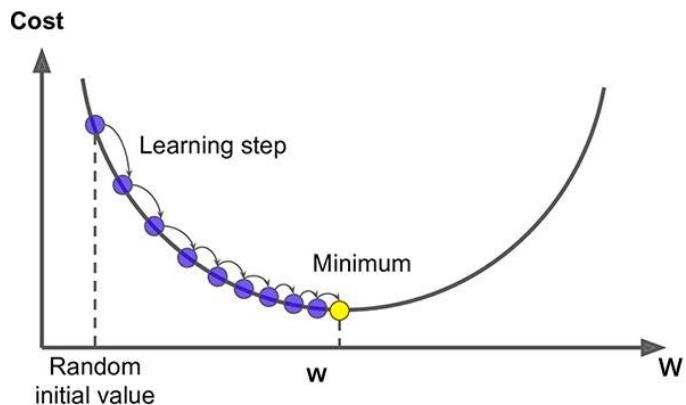
https://afonsobandeira.files.wordpress.com/2015/03/steepestdescent_compare_euclidean_sphere.png

**Optimize without stepping
off of the manifold**

Starting Point

Gradient descent on \mathbb{R}^n

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





What are the
constituent parts of
gradient descent?

Starting Point

Gradient descent on \mathbb{R}^n

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

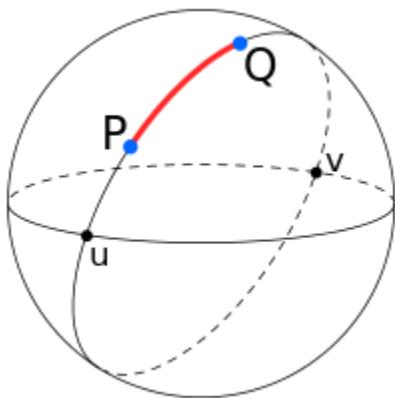
Means of walking along the domain

Gradient direction

Starting Point

Gradient descent on \mathbb{R}^n

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



Means of walking along the domain

Gradient direction

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

Manifold gradient descent (roughly)

Why Manifold Optimization?

- Practical perspective:

Better algorithms

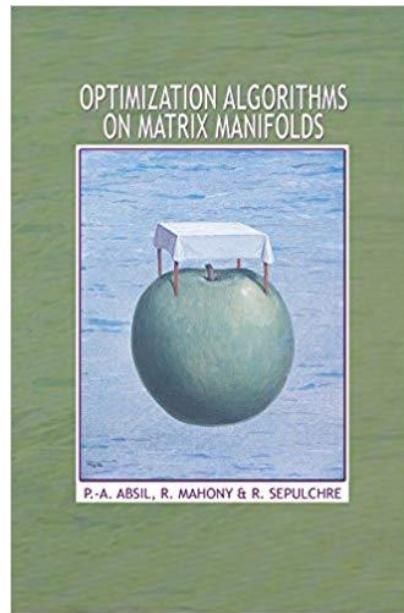
Automatic constraint satisfaction,
specialized to the space

- Theoretical perspective:

**Elegant mathematical
characterization**

Generalize convexity, gradient descent, ...

Comprehensive Introduction



Matlab Toolbox

The screenshot shows a web browser window displaying the Manopt website. The title bar reads "Manopt, a matlab toolbox for optimization on manifolds". The address bar shows "manopt.org". The page has a yellow header with navigation links: Home (selected), Tutorial, Downloads, Forum, About, and Contact. The main content area has a yellow background and features a large dark blue "Welcome to Manopt!" heading. Below it is a subtitle "A Matlab toolbox for optimization on manifolds". A paragraph explains that optimization on manifolds is a powerful paradigm for nonlinear optimization problems, mentioning Manopt's ability to handle symmetries and constraints like orthonormality and low rank. At the bottom are two buttons: "Download" and "Get started".

Manopt, a matlab toolbox for optimization on manifolds

Welcome to Manopt!

A Matlab toolbox for optimization on manifolds

Optimization on manifolds is a powerful paradigm to address nonlinear optimization problems. With Manopt, it is easy to deal with various types of symmetries and constraints which arise naturally in applications, such as orthonormality and low rank.

Download

Get started

Manifolds?

Manifolds are mathematical sets with a smooth

Key features

Manopt comes with a large library of manifolds

It's open source

Check out the license and let us know how you

Additional Resource (coming soon)

NICOLAS BOUMAL

AN INTRODUCTION TO
OPTIMIZATION ON
SMOOTH MANIFOLDS

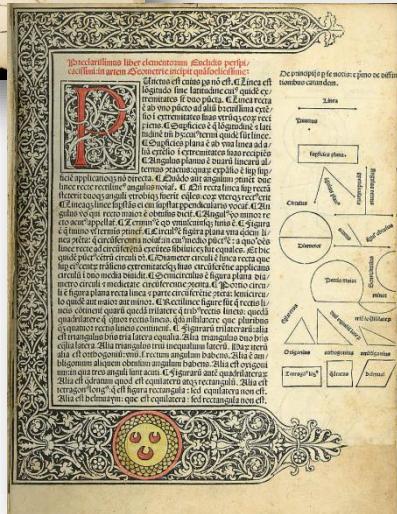
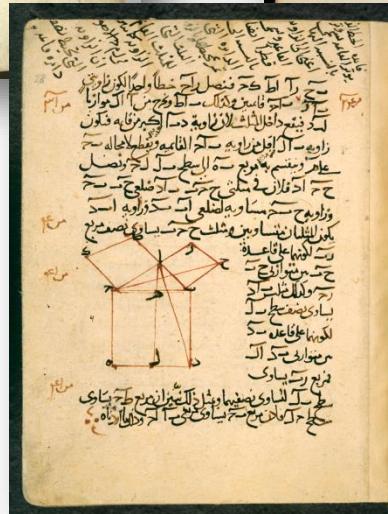
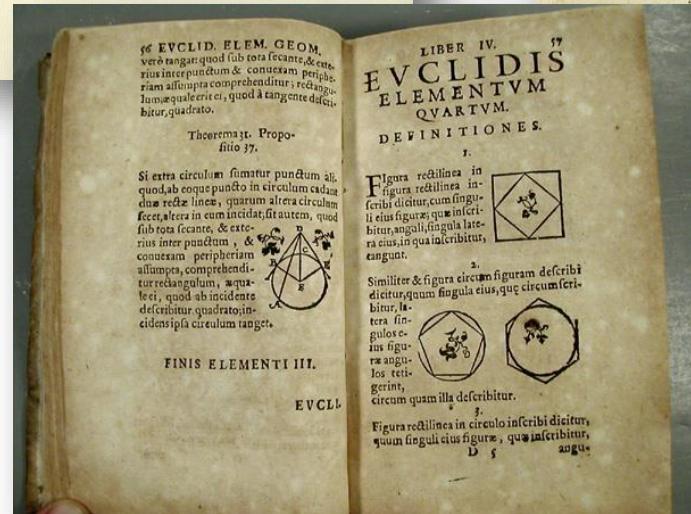
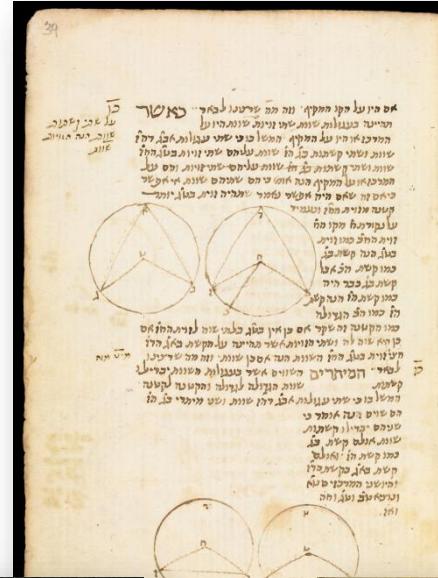
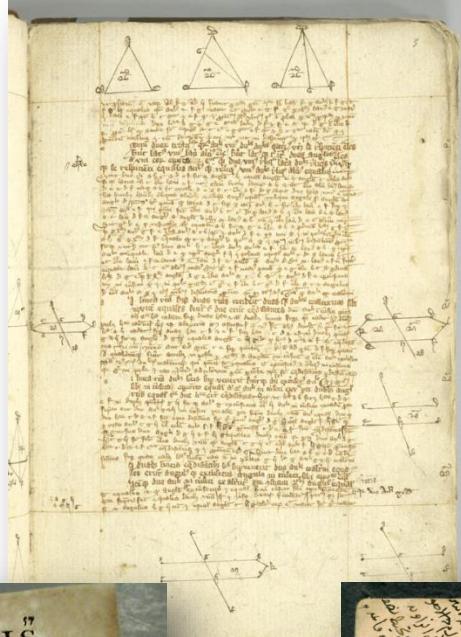
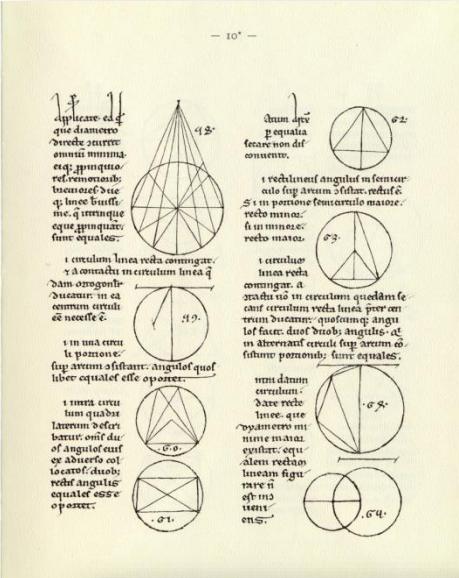
nicolasboumal.net/#book

DEPARTMENT OF MATHEMATICS, PRINCETON UNIVERSITY

Plan for Today

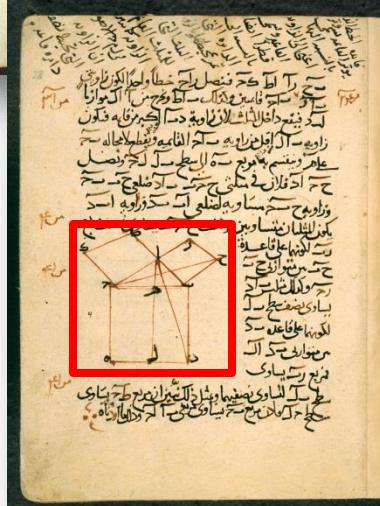
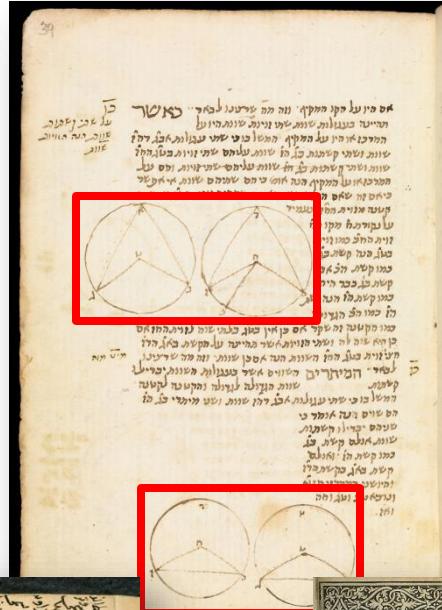
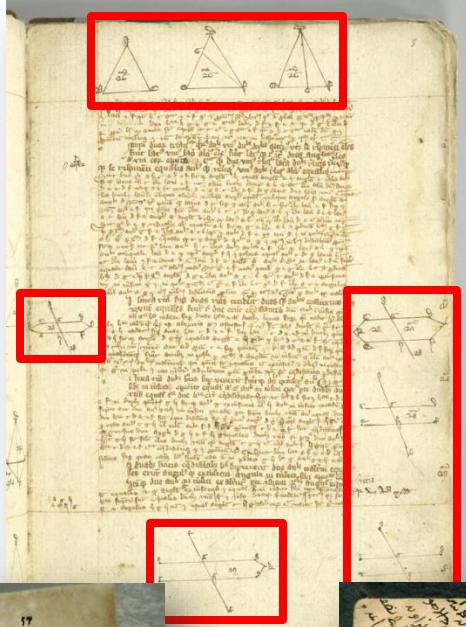
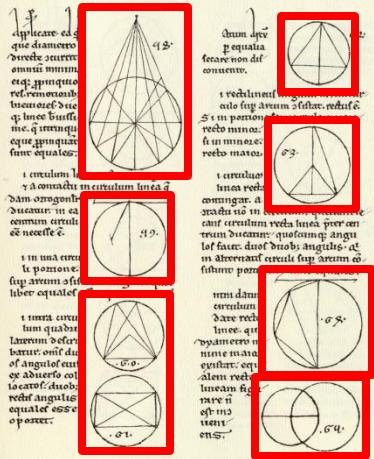
- **Landscape of geometric theories**
- **Defining a manifold**
- **Adapting gradient descent**
- **Examples**
- **What's next?**

Euclidean Geometry

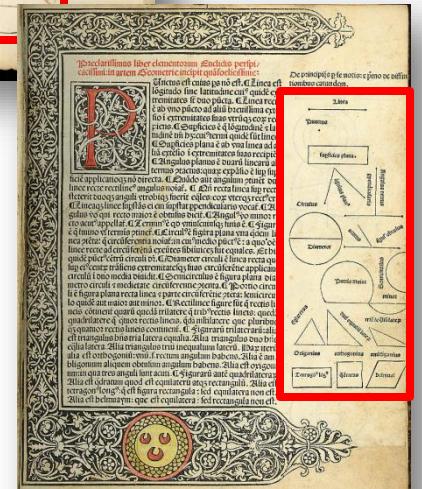
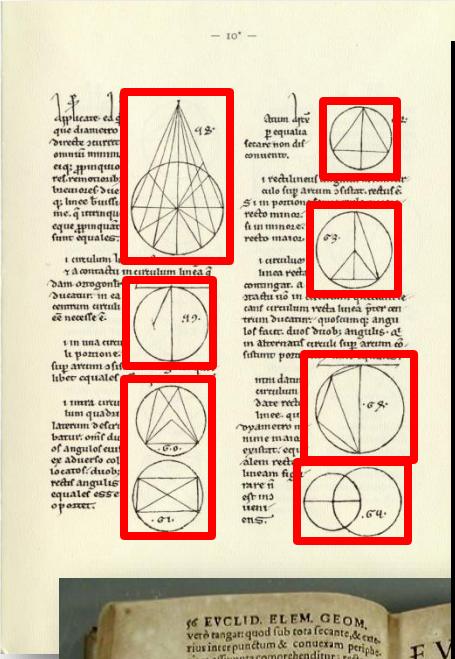


Euclidean Geometry

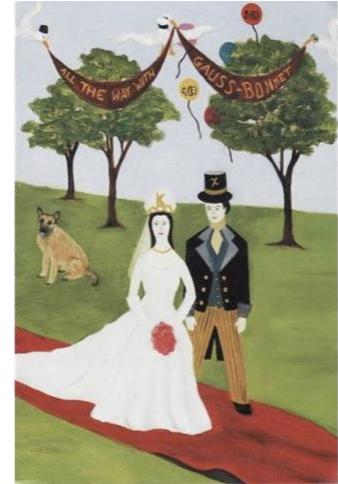
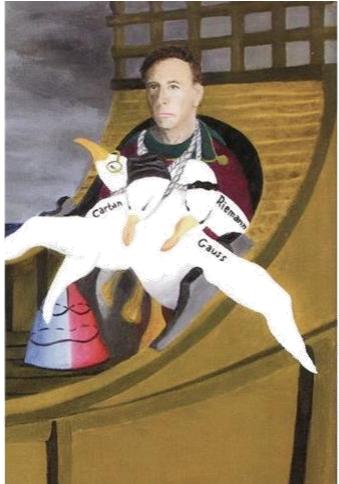
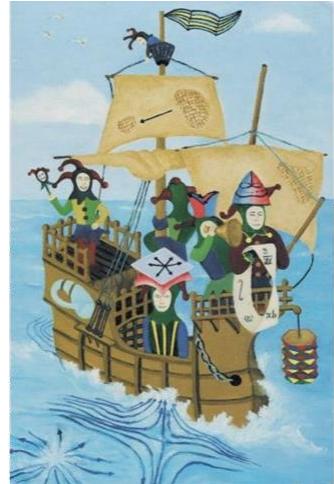
- 10* -



Euclidean Geometry

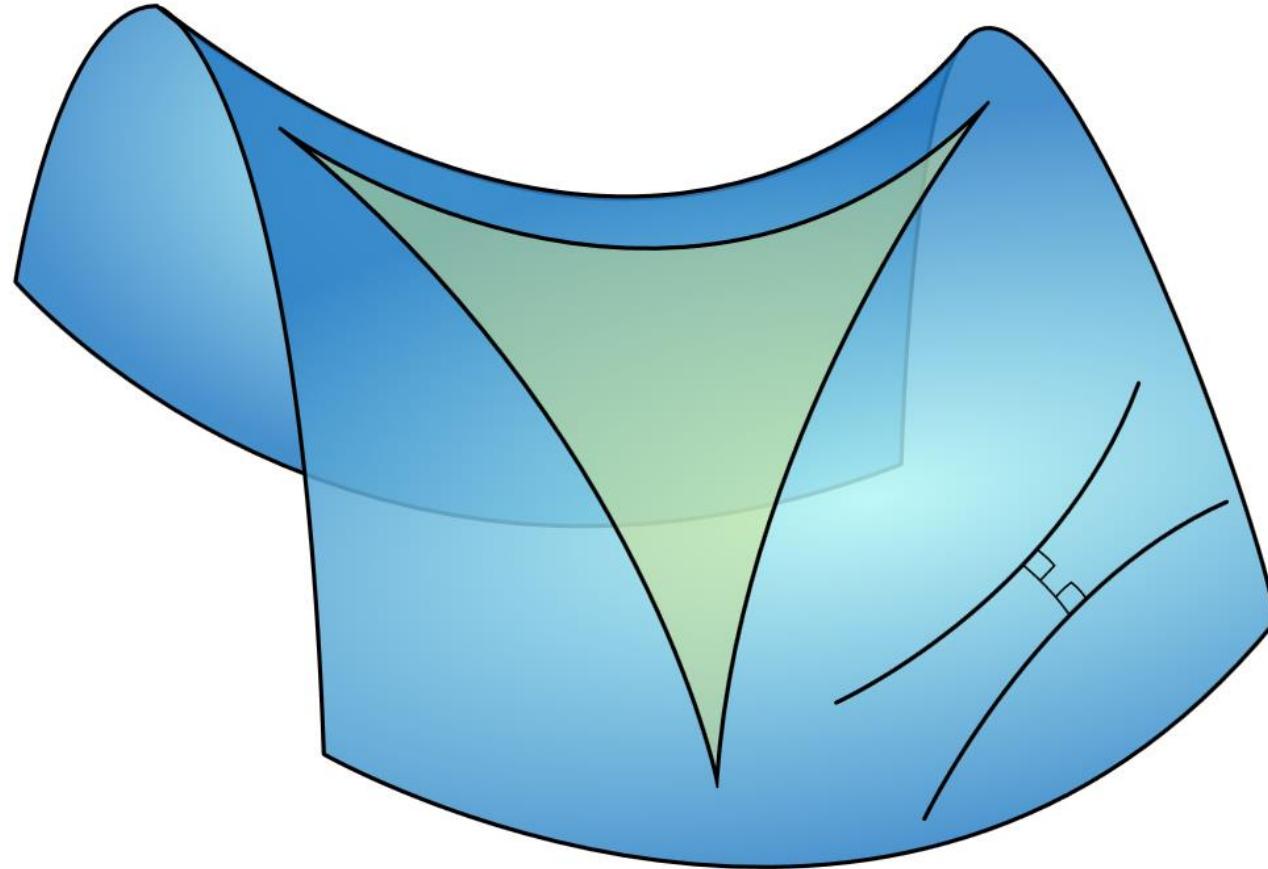


Differential Geometry



Spivak: *A Comprehensive Introduction to Differential Geometry*

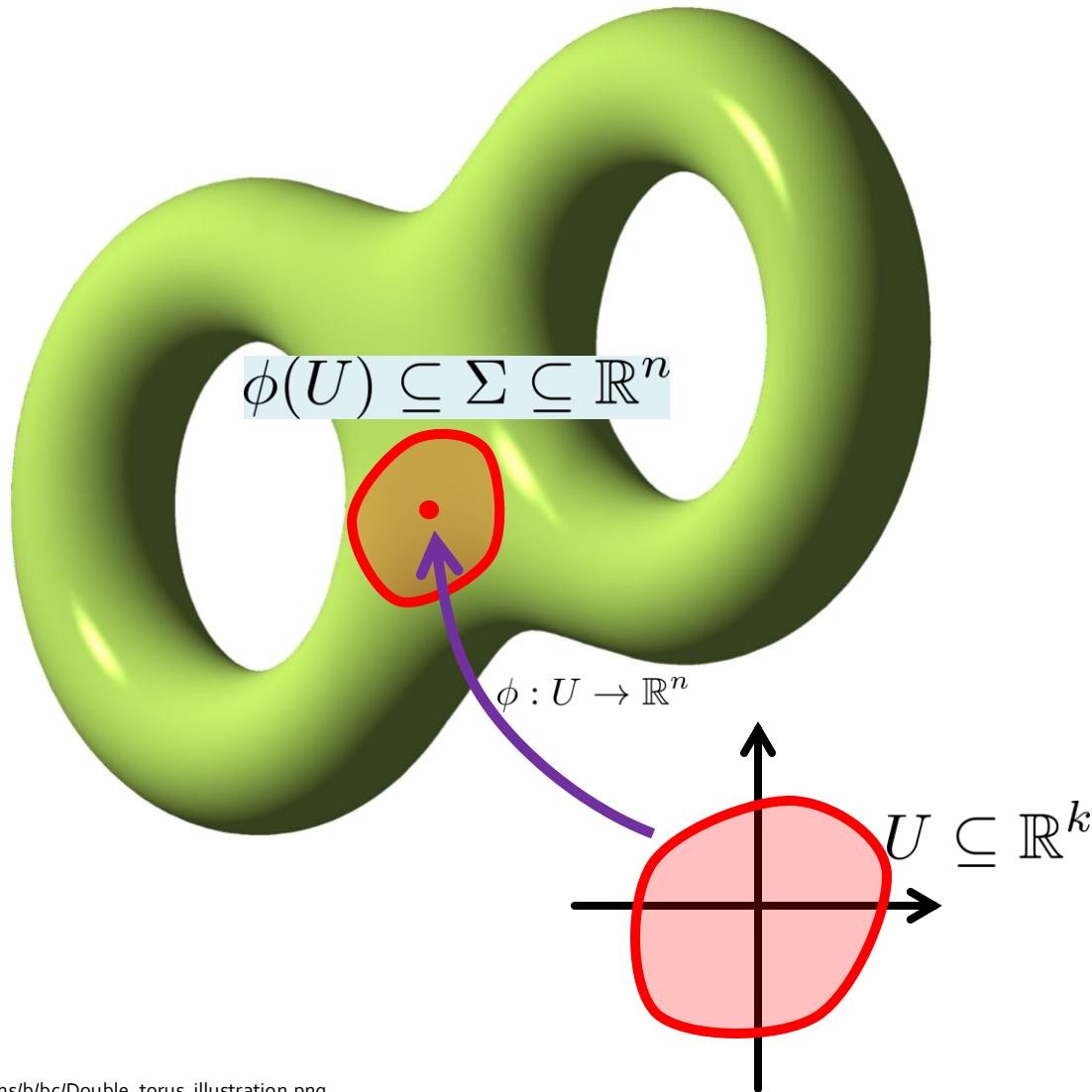
Differential Geometry



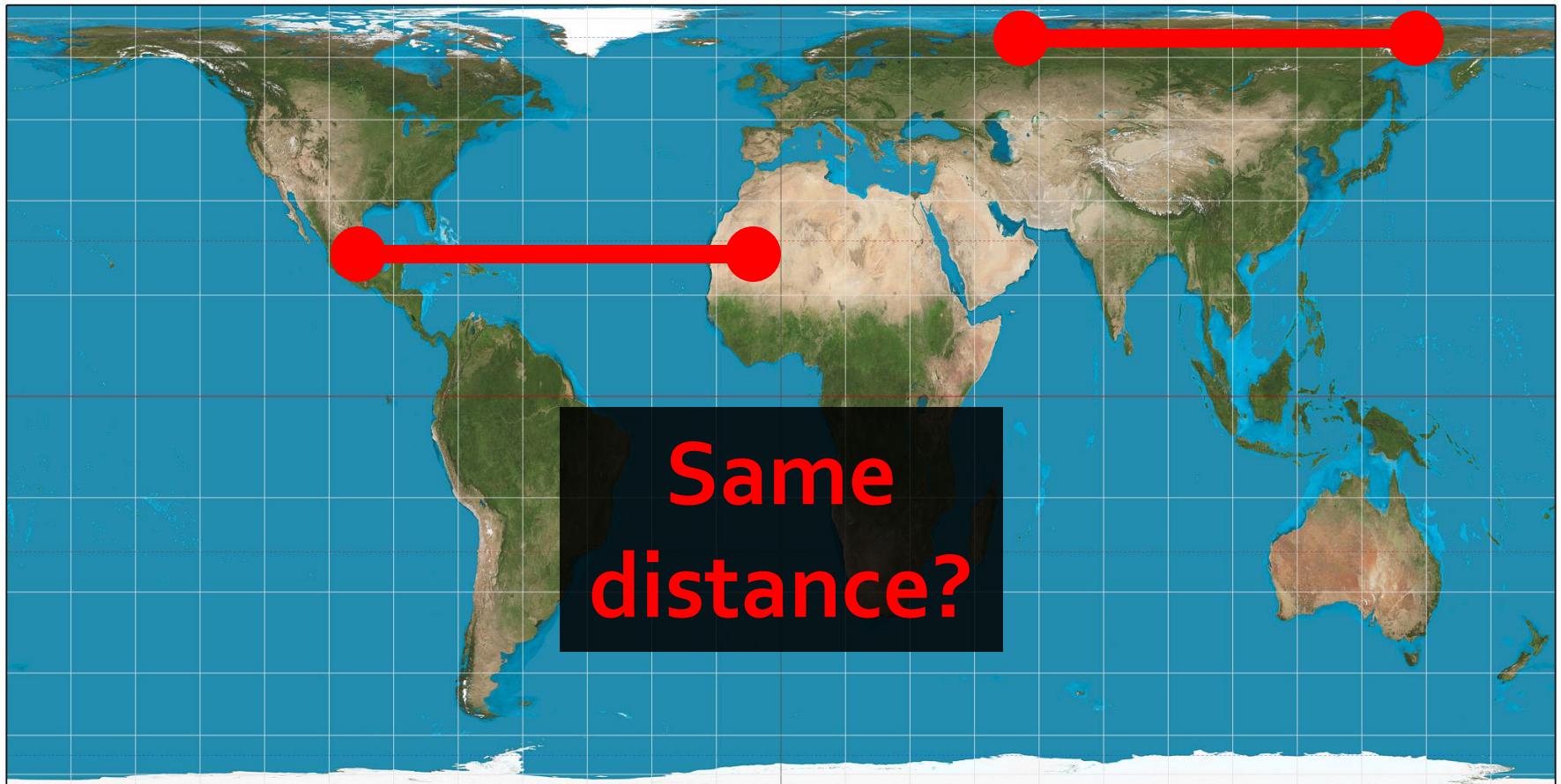
http://en.wikipedia.org/wiki/Differential_geometry

Study of smooth manifolds

Manifold



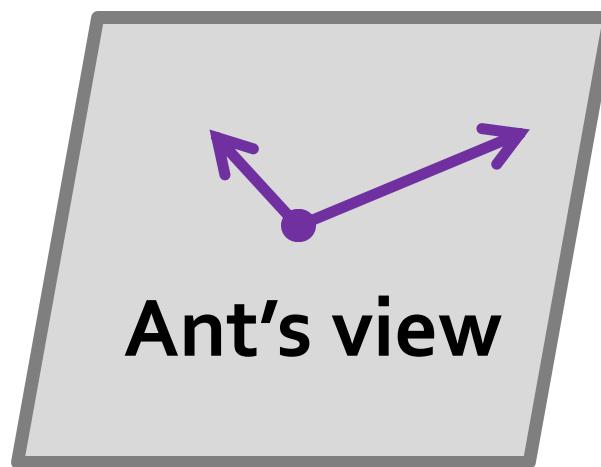
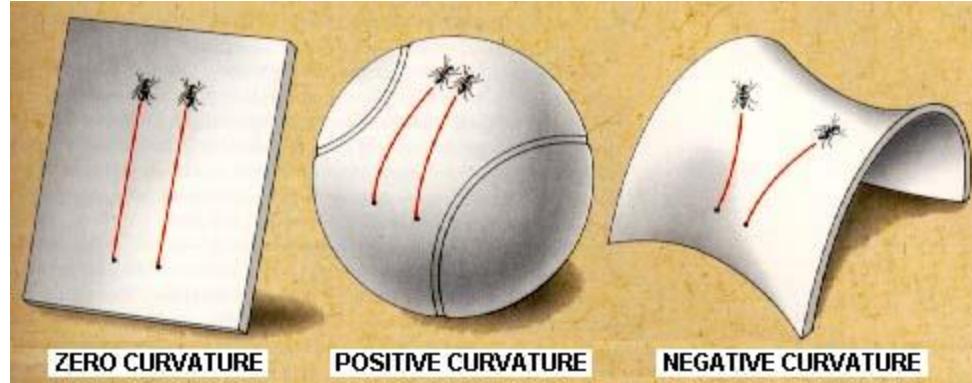
Riemannian Viewpoint



http://upload.wikimedia.org/wikipedia/commons/2/2c/Hobo%20%80%93Dyer_projection_SW.jpg

Only need angles and distances

Riemannian Viewpoint



<http://www.phy.syr.edu/courses/modules/LIGHTCONE/pics/curved.jpg>

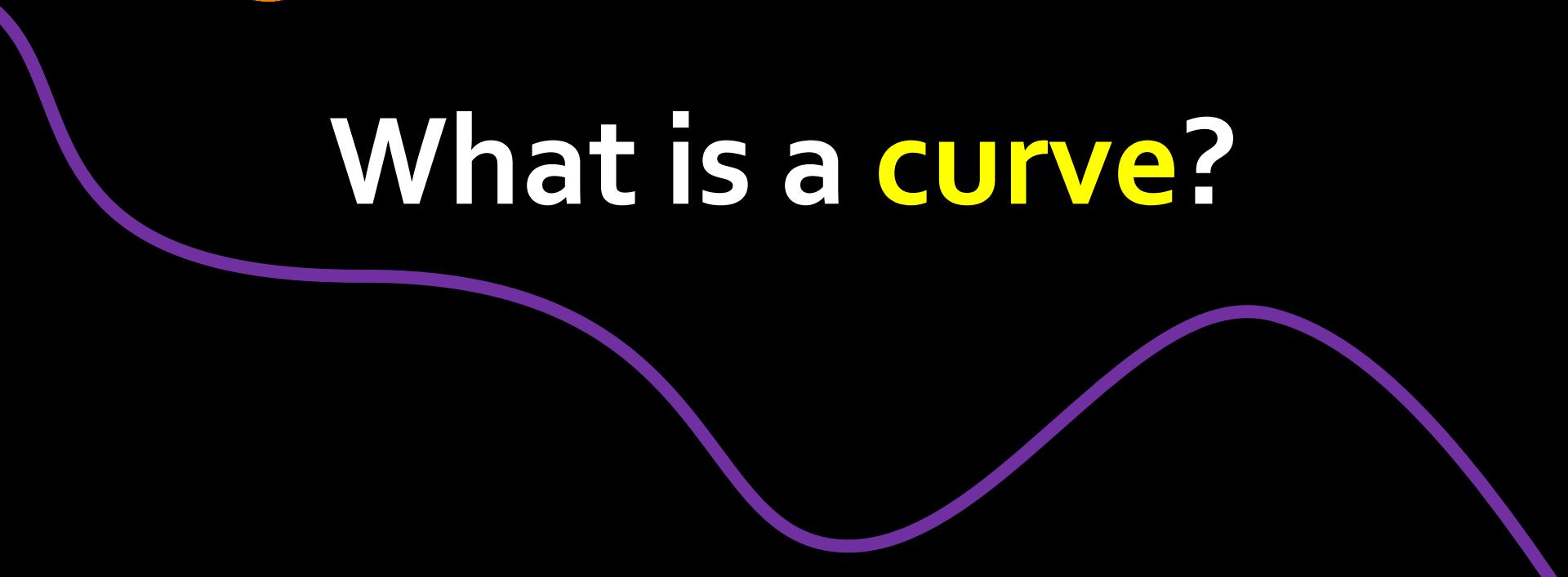
Only need angles and distances

Plan for Today

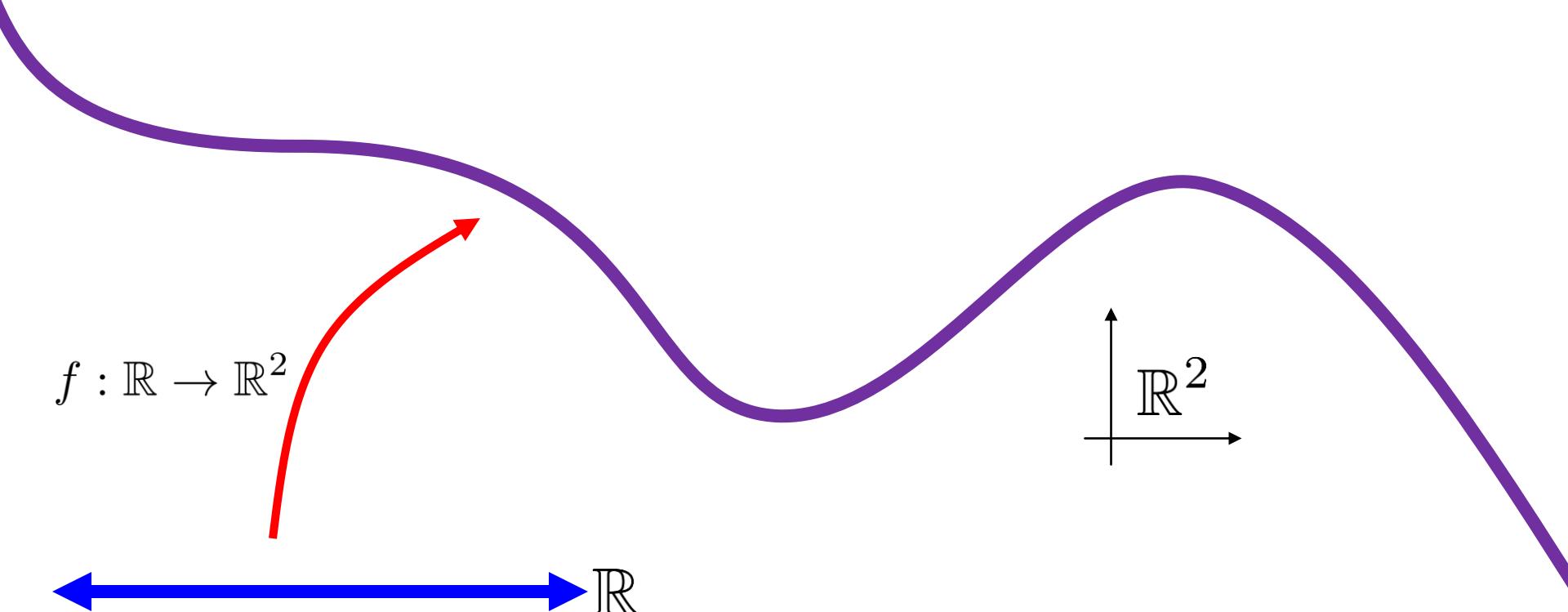
- Landscape of geometric theories
- Defining a manifold
- Adapting gradient descent
- Examples
- What's next?



What is a **curve**?



Defining “Curve”



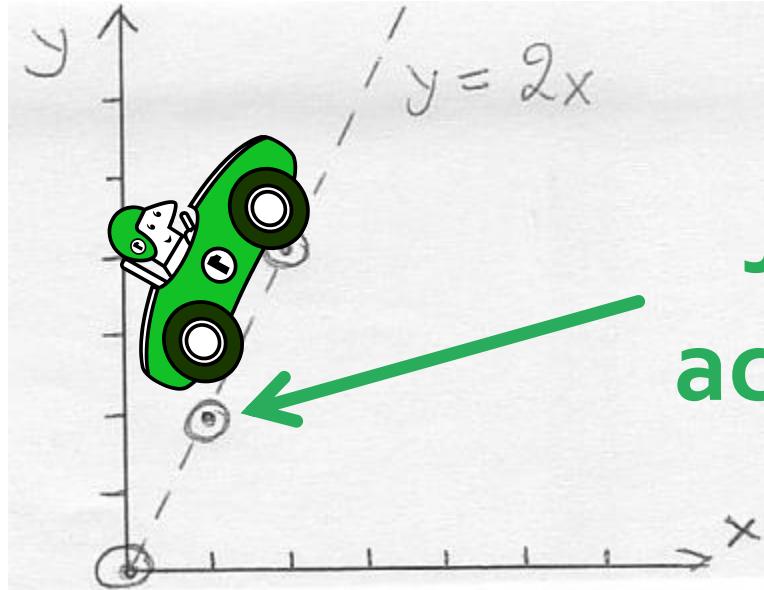
A function?

Subtlety

$$\gamma_3(t) \equiv (0, 0)$$

Not a curve

Different from Calculus

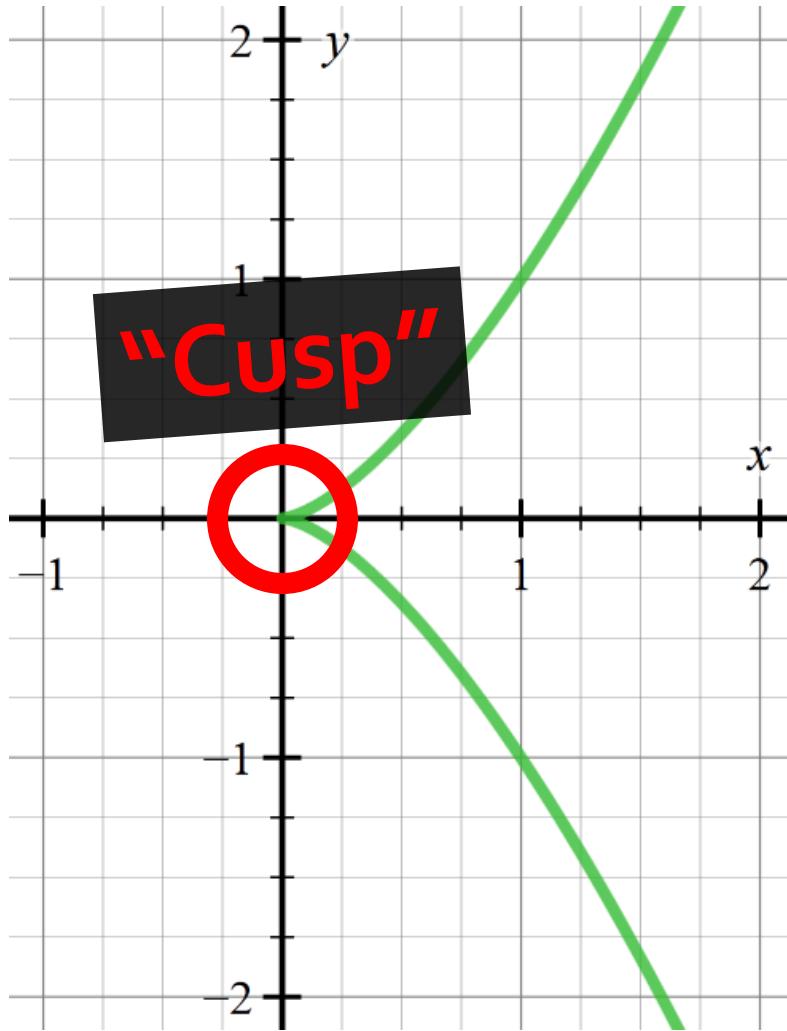


Jams on
accelerator

$$f_1(t) = (t, 2t)$$

$$f_2(t) = \begin{cases} (t, 2t) & t \leq 1 \\ (2(t - \frac{1}{2}), 4(t - \frac{1}{2})) & t > 1 \end{cases}$$

Graphs of Smooth Functions



$$f(t) = (t^2, t^3)$$

Geometry of a Curve

A curve is a
set of points
with certain properties.

It is not a function.

Geometric Definition



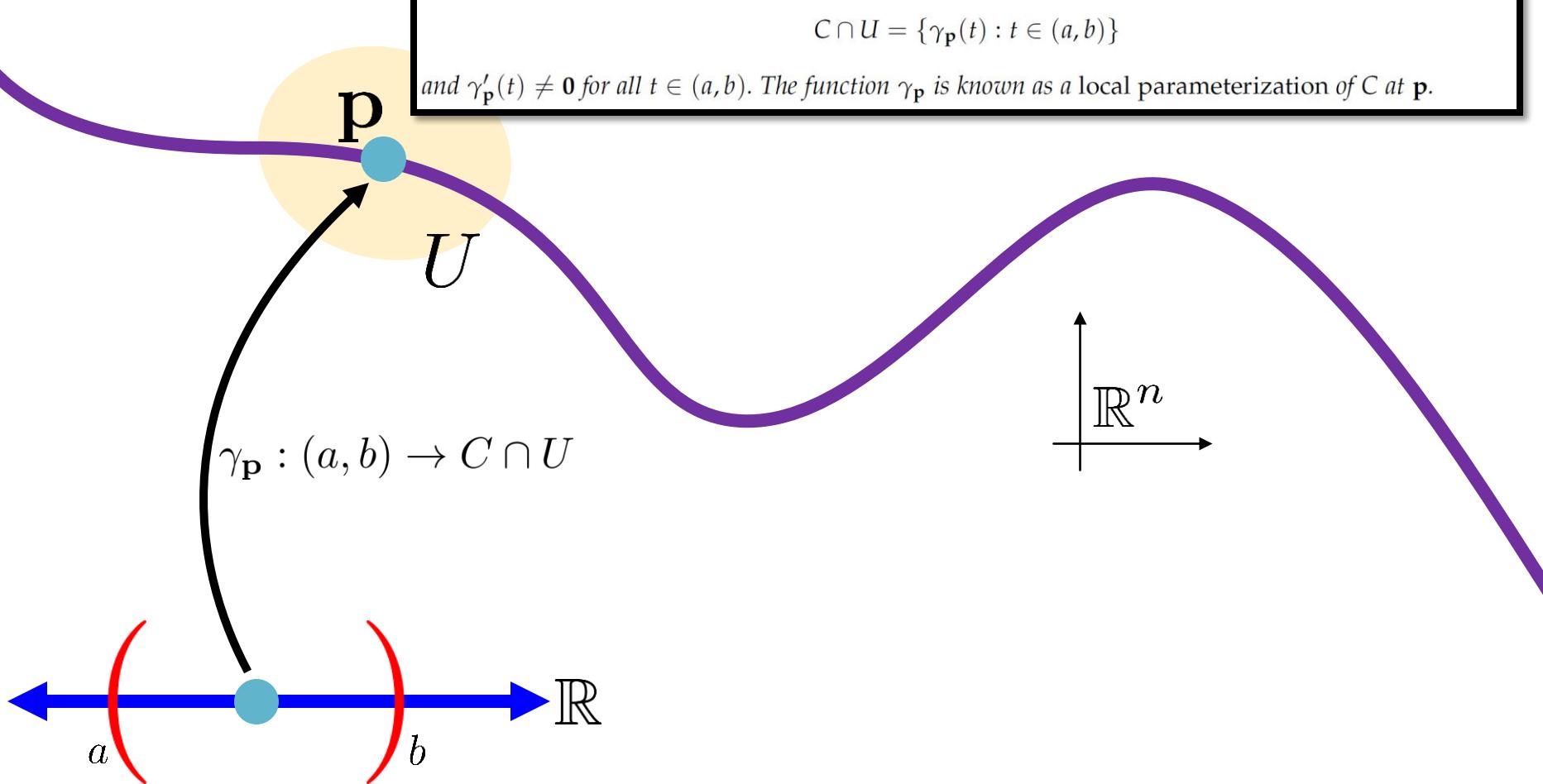
Set of points that locally looks like a line.

Formal Statement

Definition 3.1 (Differentiable curve in \mathbb{R}^n). A differentiable curve in \mathbb{R}^n is a set of points $C \subset \mathbb{R}^n$ with the property that for every $\mathbf{p} \in C$ there exists an open neighborhood $U \subseteq \mathbb{R}^n$ containing \mathbf{p} and a smooth function $\gamma_{\mathbf{p}} : (a, b) \rightarrow C \cap U$, so that

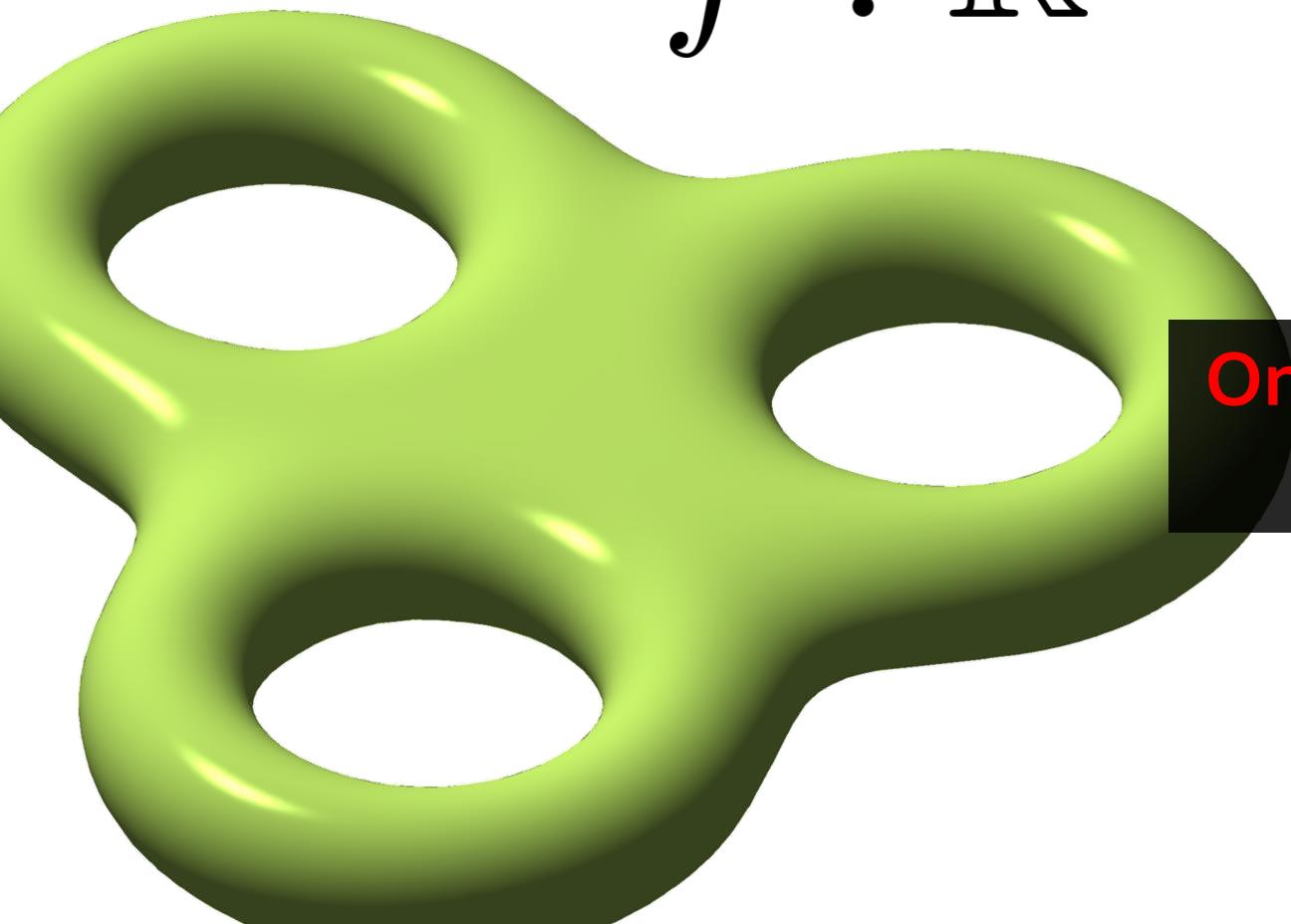
$$C \cap U = \{\gamma_{\mathbf{p}}(t) : t \in (a, b)\}$$

and $\gamma'_{\mathbf{p}}(t) \neq \mathbf{0}$ for all $t \in (a, b)$. The function $\gamma_{\mathbf{p}}$ is known as a local parameterization of C at \mathbf{p} .



Next Issue

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^3 ?$$



One function isn't
enough!

Major difference from curves!

Differential of a Function

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Matrix:

$$Df = \left(\frac{\partial f_i}{\partial x_j} \right) \in \mathbb{R}^{m \times n}$$

Linear operator:

$$df_{\mathbf{p}} : T_{\mathbf{p}} \mathbb{R}^n \rightarrow T_{f(\mathbf{p})} \mathbb{R}^m$$

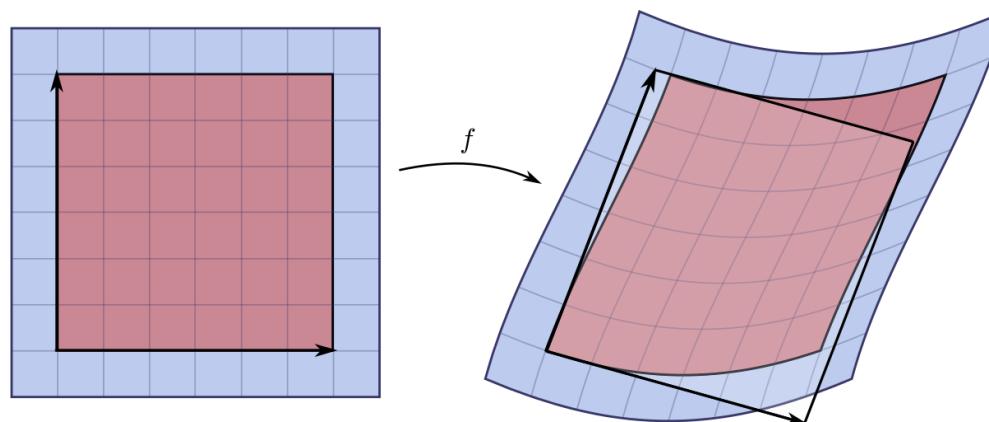
Just Like Curves

A manifold is a
set of points
with certain properties.

It is not a function.

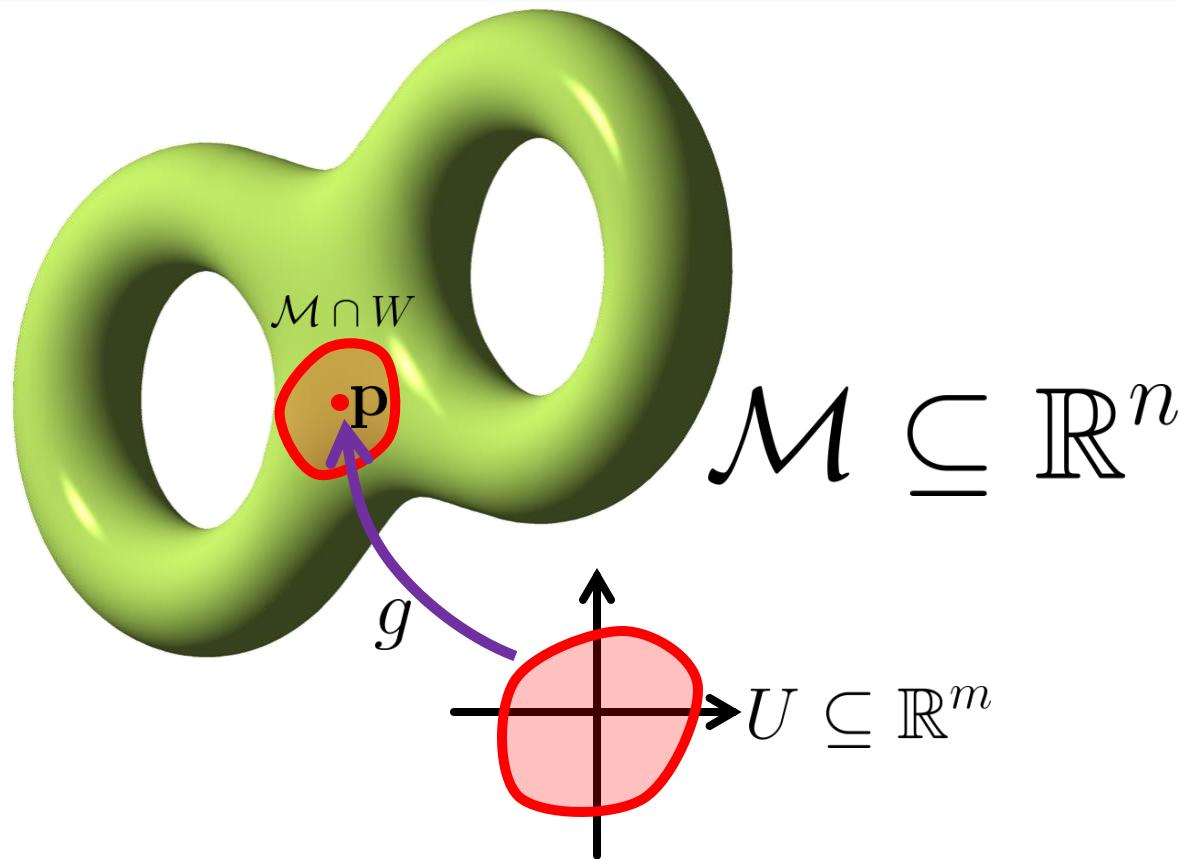
Differential Geometer's Mantra

A manifold is
locally Euclidean.



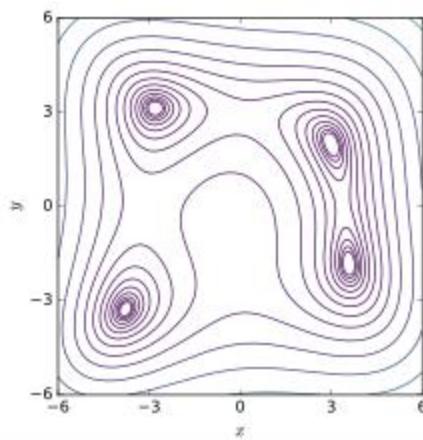
Theoretical Definition: (Sub)Manifold

Definition (Submanifold of \mathbb{R}^n , with and without boundary). A set $\mathcal{M} \subseteq \mathbb{R}^n$ is an m -dimensional submanifold of \mathbb{R}^n if for each $p \in \mathcal{M}$ there exist open sets $U \subseteq \mathbb{R}^m$, $W \subseteq \mathbb{R}^n$ and a function $g : U \cap \mathcal{H}_m \rightarrow \mathcal{M} \cap W$ such that $p \in W$ and g is a one-to-one and smooth map whose Jacobian is rank- m and admitting a continuous inverse $g^{-1} : W \cap \mathcal{M} \rightarrow U$.



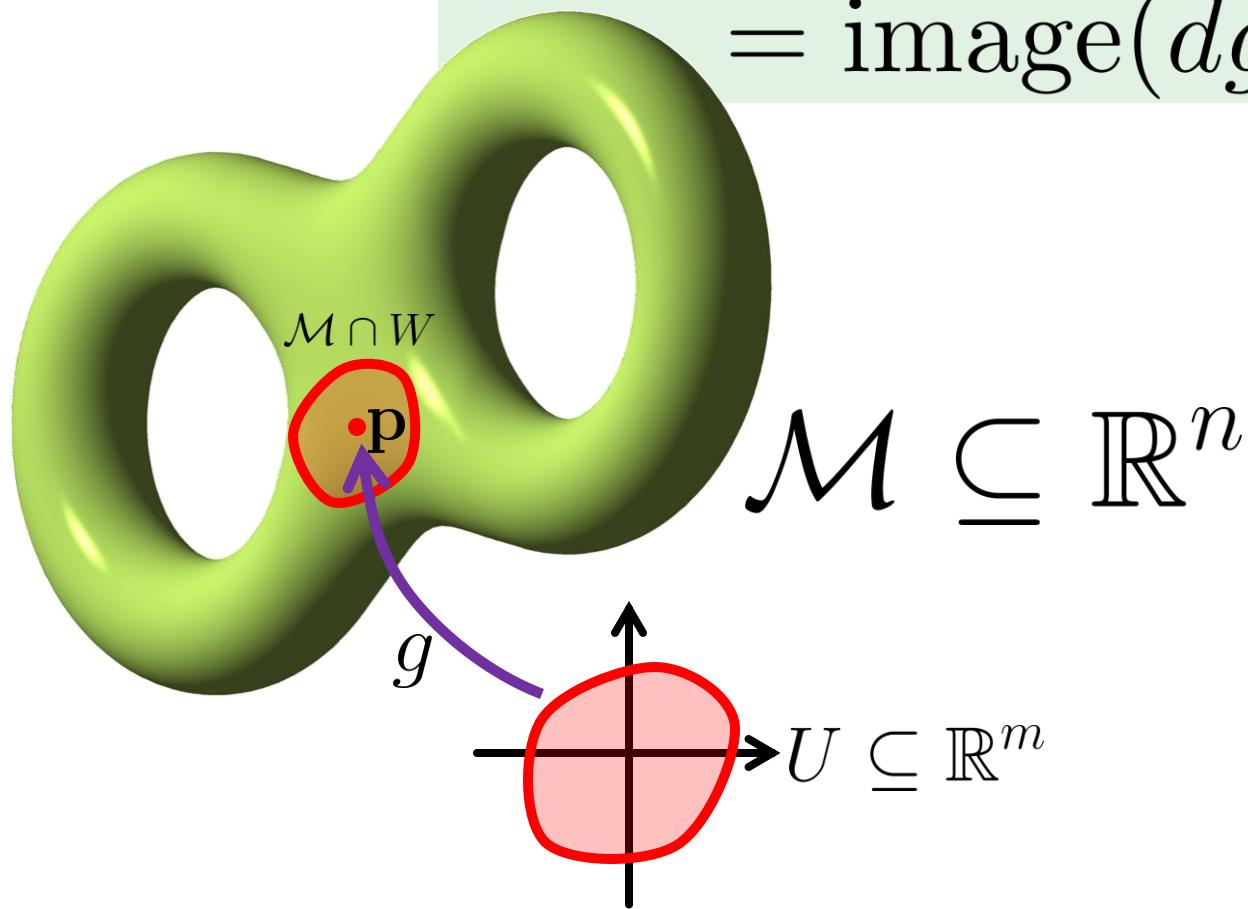
Useful Fact

Level sets $f^{-1}(c)$ of $f: \mathbb{R}^n \rightarrow \mathbb{R}$ are $(n-1)$ -dimensional manifolds as long as df is nondegenerate on $\{x \in \mathbb{R}^n : f(x) = c\}$.



Tangent Space

$$T_p \mathcal{M} = \gamma'(0), \text{ where } \gamma(0) = p \\ = \text{image}(dg_p)$$



Riemannian Inner Product

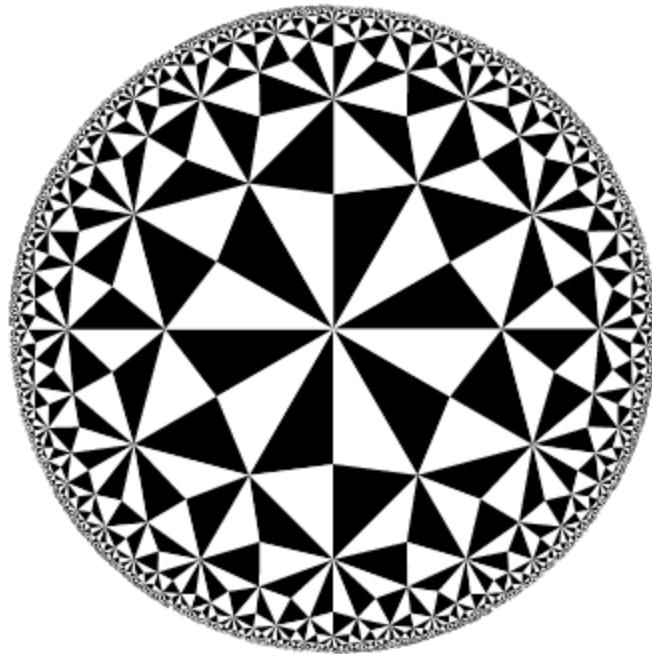
$$\langle \cdot, \cdot \rangle_p : T_p M \times T_p M \rightarrow \mathbb{R}$$



Symmetric, bilinear,
positive definite form

Example: Poincaré Disk

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



\approx



Plan for Today

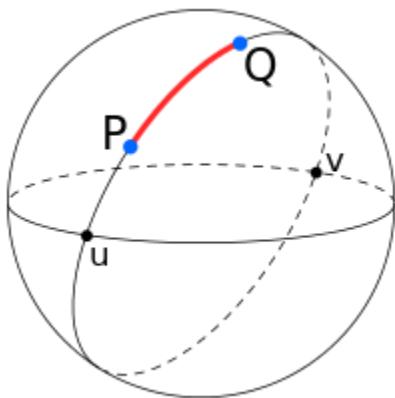
- Landscape of geometric theories
- Defining a manifold
- Adapting gradient descent
- Examples
- What's next?

Recall:

Starting Point

Gradient descent on \mathbb{R}^n

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



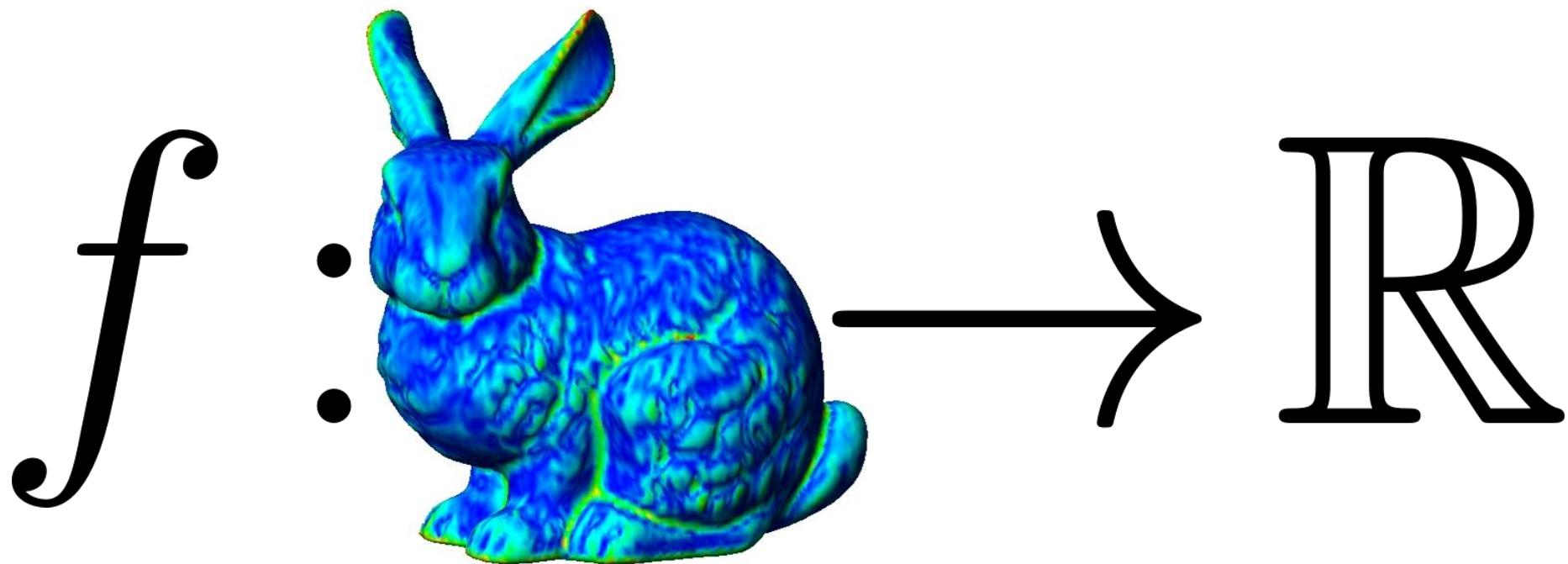
Means of walking along the domain

Gradient direction

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

Manifold gradient descent (roughly)

Scalar Functions on Manifolds



http://www.ieeta.pt/polymeco/Screenshots/PolyMeCo_OneView.jpg

Map points to real numbers

Differential of a Map

Definition (Differential). Suppose $\varphi : \mathcal{M} \rightarrow \mathcal{N}$ is a map from a submanifold $\mathcal{M} \subseteq \mathbb{R}^k$ into a submanifold $\mathcal{N} \subseteq \mathbb{R}^\ell$. Then, the differential $d\varphi_{\mathbf{p}} : T_{\mathbf{p}}\mathcal{M} \rightarrow T_{\varphi(\mathbf{p})}\mathcal{N}$ of φ at a point $\mathbf{p} \in \mathcal{M}$ is given by

$$d\varphi_{\mathbf{p}}(\mathbf{v}) := (\varphi \circ \gamma)'(0),$$

where $\gamma : (-\varepsilon, \varepsilon) \rightarrow \mathcal{M}$ is any curve with $\gamma(0) = \mathbf{p}$ and $\gamma'(0) = \mathbf{v} \in T_{\mathbf{p}}\mathcal{M}$.

Linear map of tangent spaces

$$d\varphi_{\mathbf{p}}(\gamma'(0)) := (\varphi \circ \gamma)'(0)$$

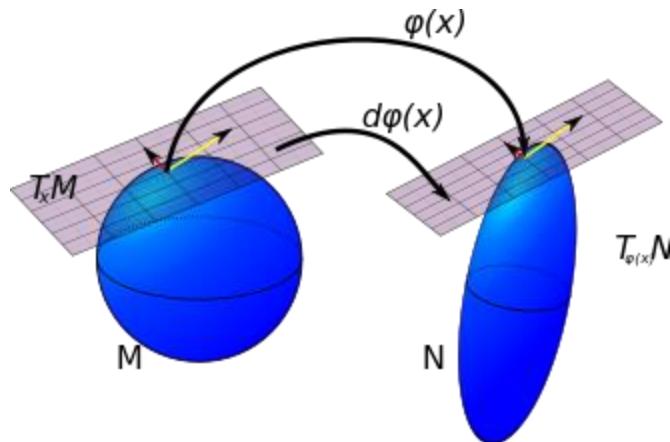
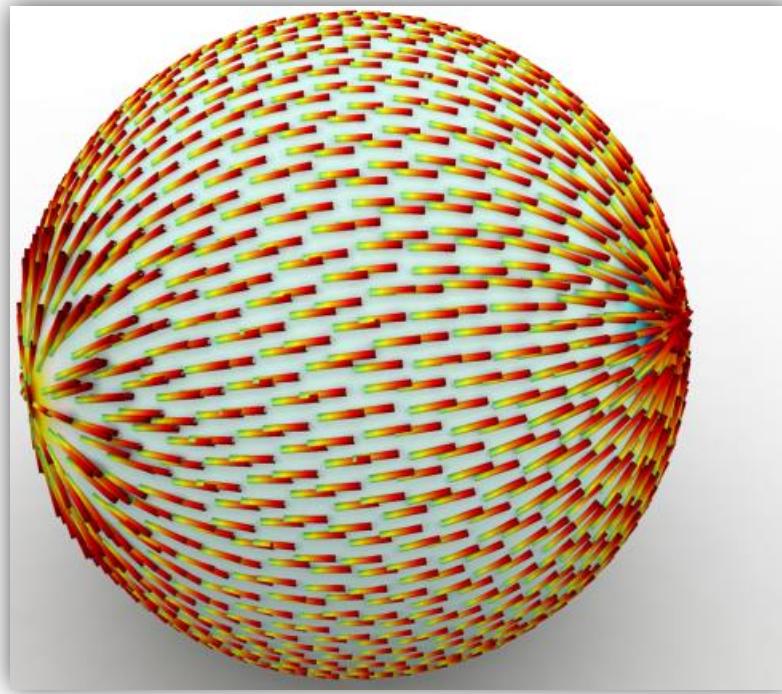


Image from Wikipedia

Gradient

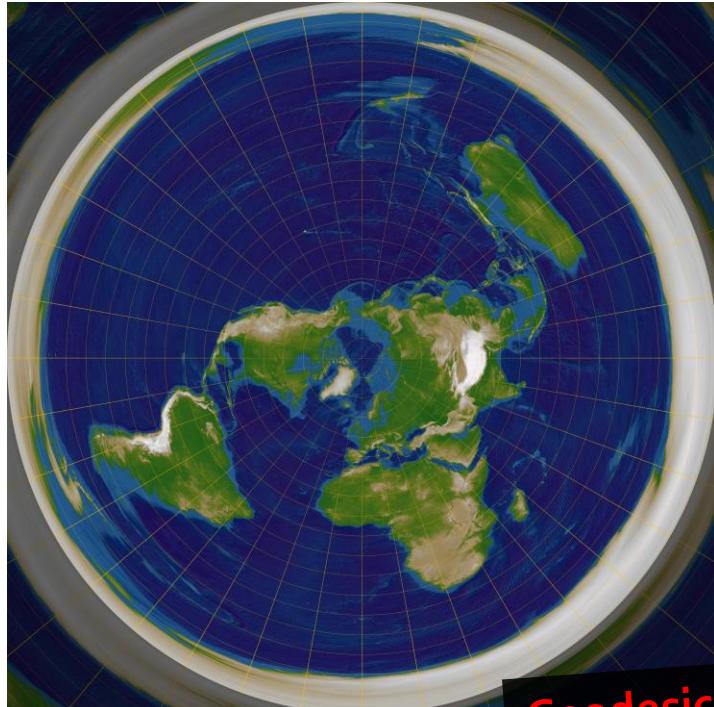
$$x_{k+1} = \exp_{x_k}(-\alpha_k \nabla f(x_k))$$



Proposition 9.2. 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

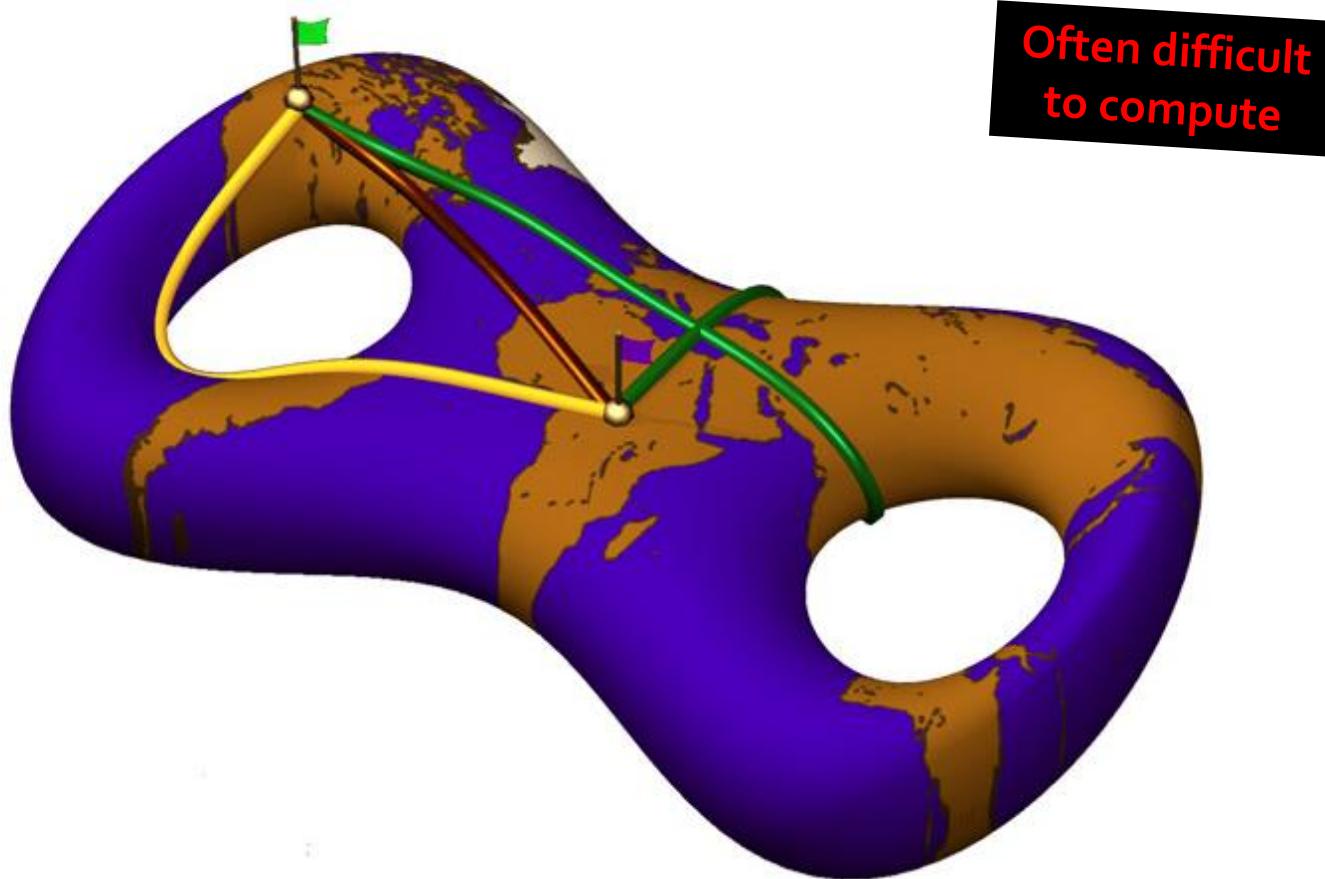
$$x_{k+1} = \exp_{x_k}(-\alpha_k \nabla f(x_k))$$



$$\exp_p(v) := \gamma_v(1)$$

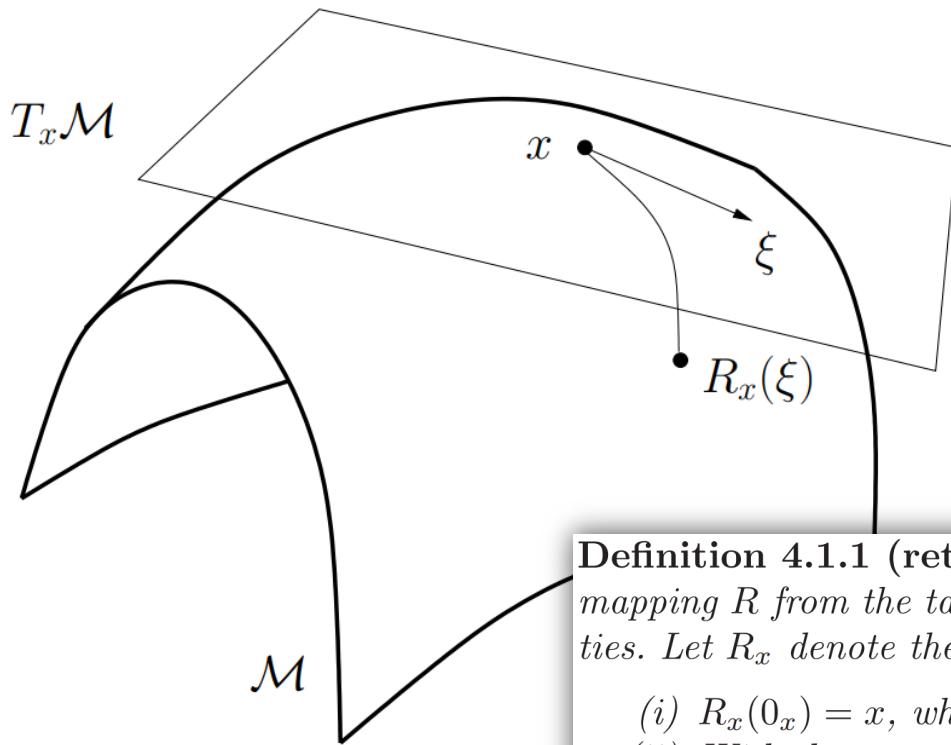
$\gamma_v(1)$ where γ_v is
(unique) geodesic from p
with velocity v .

Geodesics are Complicated!



Weaker Notion: Retraction

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



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

$$DR_x(0_x) = \text{id}_{T_x\mathcal{M}},$$

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

Extension: Riemannian Gradient

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

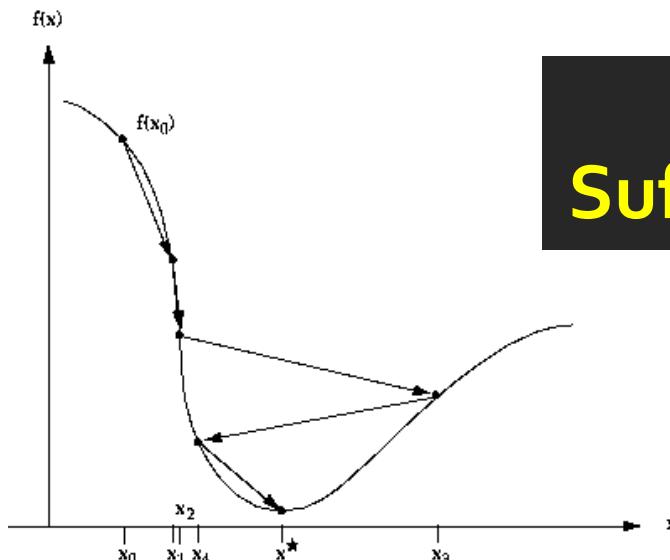
$$\Rightarrow \nabla_g f = g^{-1} \nabla f$$

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

Extension: Newton's Method?

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



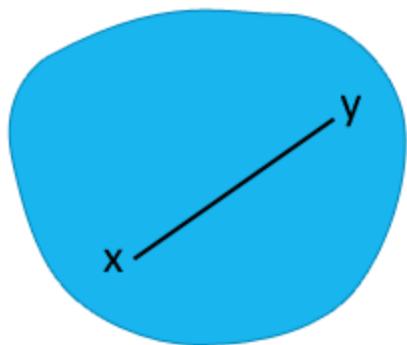
Tough to define

On manifolds gives a different search direction

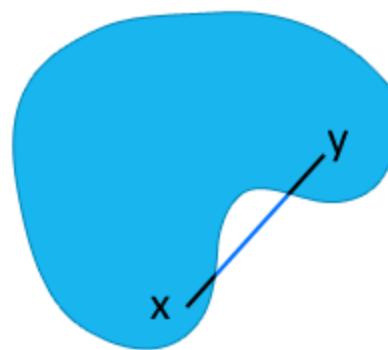
<omit>

Extension: Geodesic Convexity

Convex set



Non - convex set

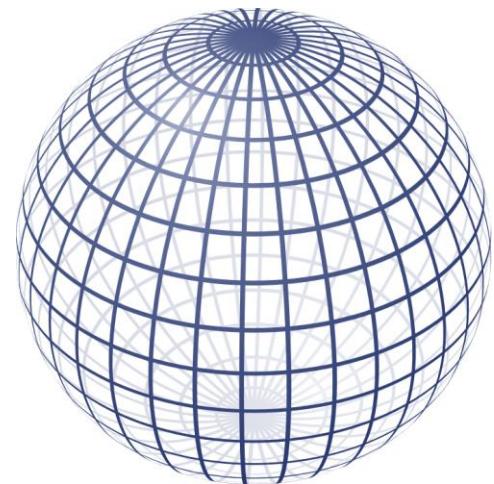


<omit>

Plan for Today

- Landscape of geometric theories
- Defining a manifold
- Adapting gradient descent
- Examples
- What's next?

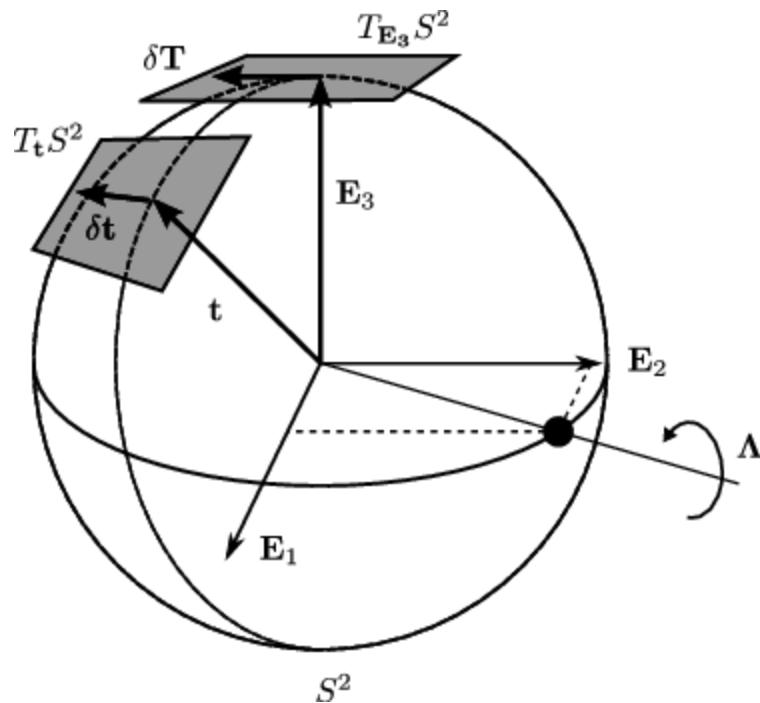
Example: Unit Sphere



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

Tangent Space of Sphere

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



Gradient

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

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

Project ambient gradient into the tangent plane

Retraction: Two (Typical) Options

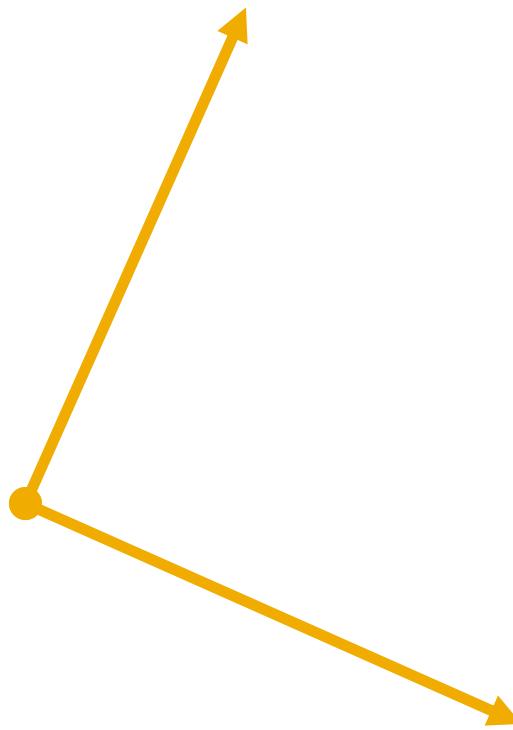
- Exponential map

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

- Projection

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

Example: Stiefel Manifold



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

Tangent Space and Retraction

To motivate on the board:

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

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

Plan for Today

- Landscape of geometric theories
- Defining a manifold
- Adapting gradient descent
- Examples
- What's next?

Rayleigh Quotient Minimization

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

Assume A is symmetric

On the board:

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

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

Plan for Today

- Landscape of geometric theories
- Defining a manifold
- Adapting gradient descent
- Examples
- 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)	$\sum_{i,j} (d_X(x_i, x_j) - d_Y(M^\top x_i, M^\top x_j))^2$	$\mathcal{O}^{d \times r}$	$M^\top X$
LDA (§3.1.3)	$\frac{\text{tr}(M^\top \Sigma_B M)}{\text{tr}(M^\top \Sigma_W M)}$	$\mathcal{O}^{d \times r}$	$M^\top X$
Traditional CCA (§3.1.4)	$\text{tr}\left(M_a^\top (X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2} M_b\right)$	$\mathcal{O}^{d_a \times r} \times \mathcal{O}^{d_b \times r}$	$M_a^\top (X_a X_a^\top)^{-1/2} X_a,$ $M_b^\top (X_b X_b^\top)^{-1/2} X_b$
Orthogonal CCA (§3.1.4)	$\frac{\text{tr}\left(M_a^\top X_a X_b^\top M_b\right)}{\sqrt{\text{tr}(M_a^\top X_a X_a^\top M_a) \text{tr}(M_b^\top X_b X_b^\top M_b)}}$	$\mathcal{O}^{d_a \times r} \times \mathcal{O}^{d_b \times r}$	$M_a^\top X_a, M_b^\top X_b$
MAF (§3.1.5)	$\frac{\text{tr}(M^\top \Sigma_\delta M)}{\text{tr}(M^\top \Sigma M)}$	$\mathcal{O}^{d \times r}$	$M^\top X$
SFA (§3.1.6)	$\text{tr}(M^\top \dot{X} \dot{X}^\top M)$	$\mathcal{O}^{d \times r}$	$M^\top X$
SDR (§3.1.7)	$\text{tr}\left(\bar{K}_Z (\bar{K}_{M^\top X} + n\epsilon I)^{-1}\right)$	$\mathcal{O}^{d \times r}$	$M^\top X$
LPP (§3.1.8)	$\text{tr}(M^\top (XDX^\top)^{-1/2} X L X^\top (XDX^\top)^{-1/2} M)$	$\mathcal{O}^{d \times r}$	$M^\top (XDX^\top)^{-1/2} X$
UICA (§3.2.1)	$\frac{1}{2} \log M^\top M + \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^r \log f_\theta(m_k^\top x_n)$	$\mathbb{R}^{d \times r}$	$M^\top X$
PPCA (§3.2.2)	$\log MM^\top + \sigma^2 I + \text{tr}(XX^\top (MM^\top + \sigma^2 I)^{-1})$	$\mathbb{R}^{d \times r}$	$M^\top (MM^\top + \sigma^2 I)^{-1} X$
FA (§3.2.3)	$\log MM^\top + D + \text{tr}(XX^\top (MM^\top + D)^{-1})$	$\mathbb{R}^{d \times r}$	$M^\top (MM^\top + D)^{-1} X$
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)	$\sum_{i,j \in \eta(i)} \left\{ d_M(x_i, x_j)^2 + \lambda \sum_\ell \mathbf{1}(z_i \neq z_\ell) \right. \\ \left. [1 + d_M(x_i, x_j)^2 - d_M(x_i, x_\ell)^2]_+ \right\}$	$\mathbb{R}^{d \times r}$	$M^\top X$

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

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 2002

Abstract. In this paper, we present a nonlinear programming (SDPs) in standard form. The algorithm's distinguishing feature is the use of a low-rank factorization of the symmetric, positive semidefinite variable X of the SDP with a rank constraint $\text{rank}(X) \leq r$. The rank of the factorization, i.e., the number of columns in R , is determined by the user to enhance computational speed while maintaining equivalence of the factorized problem to the original one. Theorems concerning the convergence of the algorithm are derived, and encouraging numerical results on several test problems are also presented.

Key words. semidefinite programming – low-rank factorization – alternating direction method of multipliers – Lagrangian – limited memory BFGS

1. Introduction

In the past few years, the topic of semidefinite programming has received considerable attention in the optimization community. This is due to the fact that semidefinite programming has included the investigation of theoretically efficient algorithms, the development of practical implementation codes, and the exploration of applications in various fields.

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 optimization problem with many local minima. We show that under appropriate conditions, the global minimum of the Burer–Monteiro factorization is unique and corresponds to the global minimum of the original SDP. Our analysis is based on a deterministic guarantee for the Burer–Monteiro factorization of smooth semidefinite programs.

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

Low-Rank Completion

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

© 2013 Society for Industrial and Applied Mathematics

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 $\{1, \dots, m\} \times \{1, \dots, n\}$. The low-rank matrix completion problem [16] consists of finding the matrix with lowest rank that agrees with A on Ω :

Synchronization

Article



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

The International Journal of
Robotics Research
2019, Vol. 38(2-3) 95–125
© The Author(s) 2018
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0278364918784361
journals.sagepub.com/home/ijr



David M Rosen¹, Luca Carlone², Afonso S Bandeira³, and John J Leonard⁴

Abstract

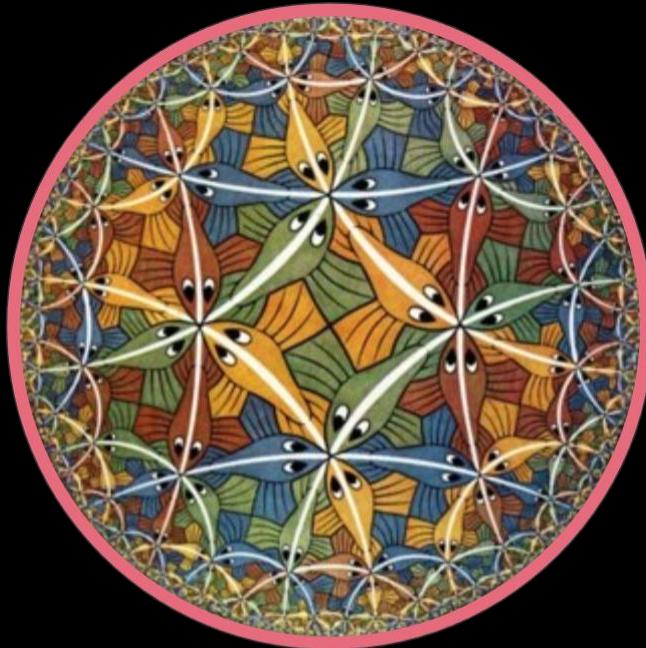
Many important geometric estimation problems naturally take the form of synchronization over the special Euclidean group: estimate the values of a set of unknown group elements $x_1, \dots, x_n \in \text{SE}(d)$ given noisy measurements of a subset of their pairwise relative transforms $x_i^{-1}x_j$. 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

The screenshot shows a GitHub browser interface for the repository `NicolasBoumal / manopt`. The current branch is `master`. The `Code` tab is selected. The `examples` directory is expanded, showing several MATLAB files with their descriptions:

- `PCA_stochastic.m`: New example PCA_stochastic to go with stochasticgra...
- `dominant_invariant_subspace.m`: Added a few comments regarding implementation of...
- `dominant_invariant_subspace_comp...`: Made a complex version of the dominant subspace ex...
- `doubly_stochastic_denoising.m`: Cosmetics
- `elliptope_SDP.m`: cosmetic
- `elliptope_SDP_complex.m`: New example: elliptope SDP solver for complex SDP's
- `essential_svd.m`: Reverting some changes of commit bc37298

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



Optimization on Manifolds

Justin Solomon
MLSS 2019

