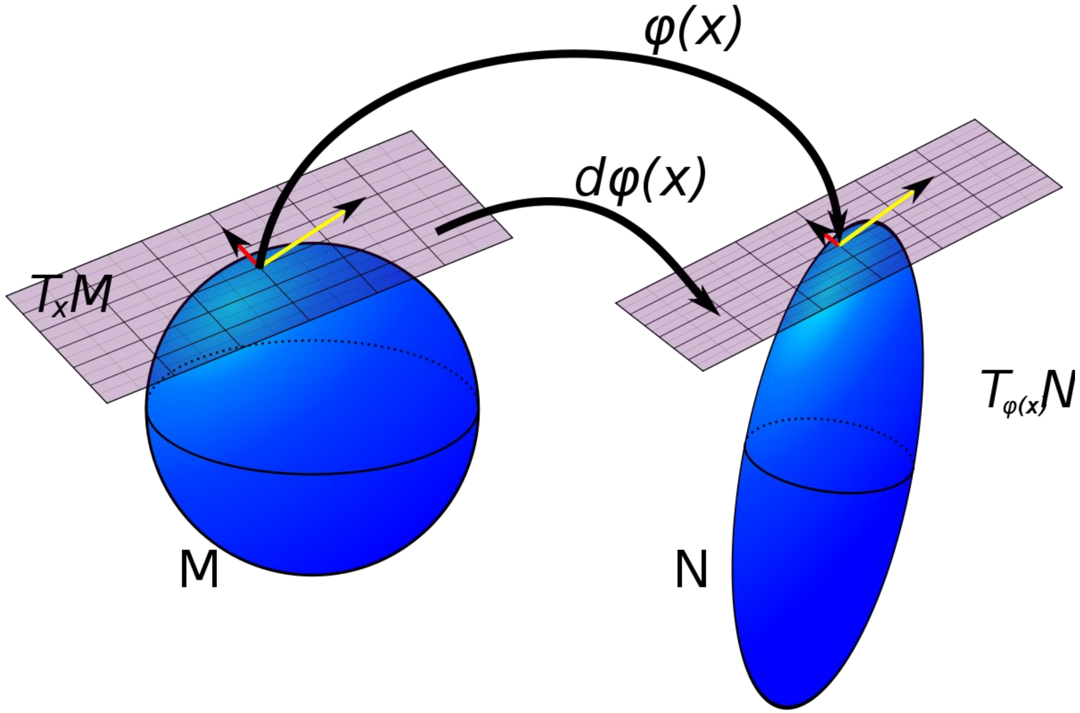
EECE 5550: Mobile Robotics



Lecture 2: An Introduction to Manifolds and Lie Groups

Motivation

Question: What is the "shape" of SO(2)?

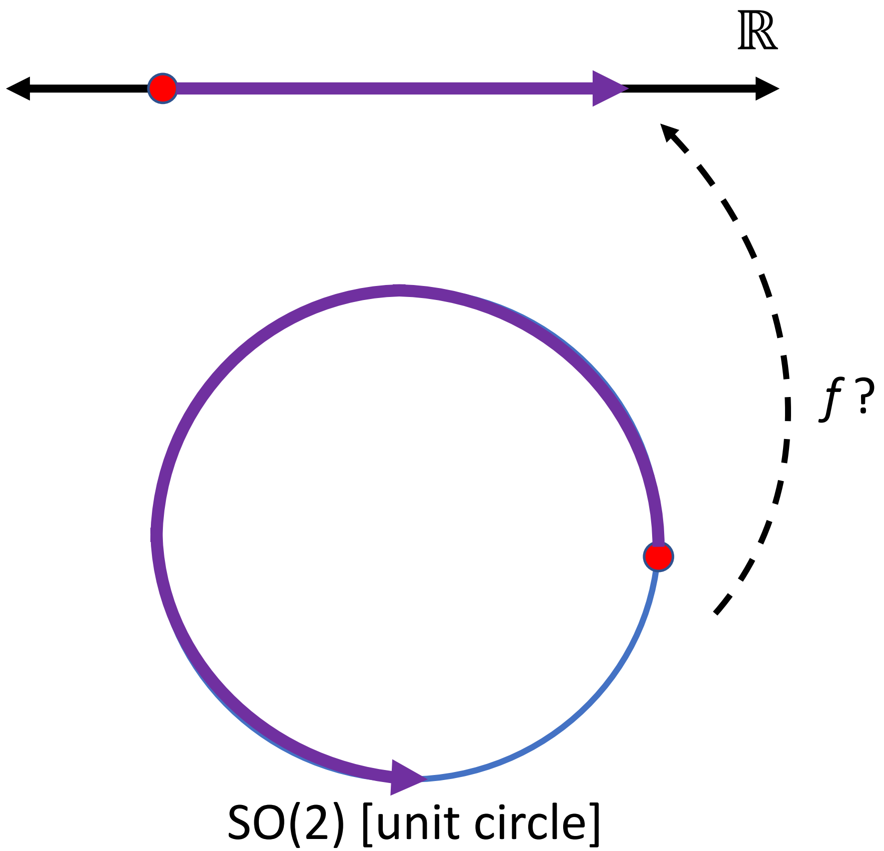
NB: We often *represent* 2D rotations using a rotation *angle* $\theta \in \mathbb{R}$.

Key question: Is it possible to do this *consistently*?

That is: is there a *continuous* mapping $f : SO(2) \to \mathbb{R}$ that will associate an *angle* with each *rotation*?

Answer: NO! (What happens when we "close the loop"?)

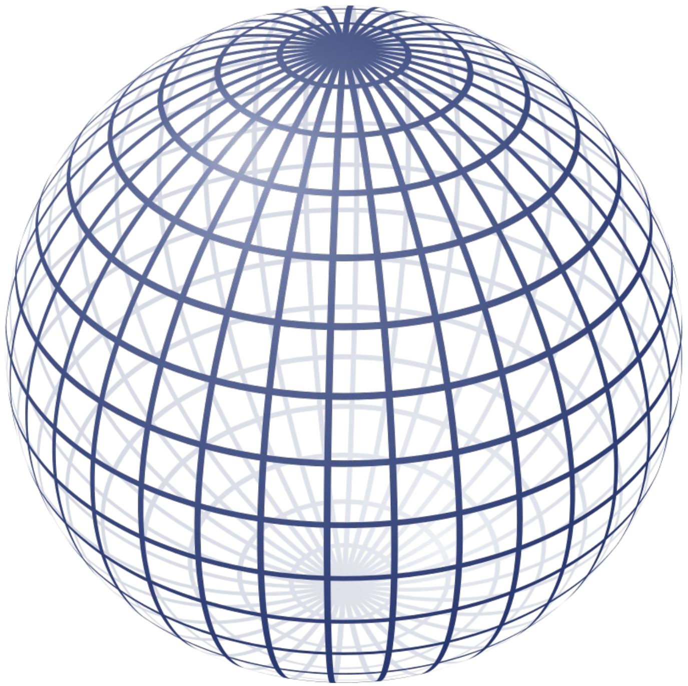
The Fundamental Problem: SO(2) is not $\mathbb{R}^n$!



Manifolds to the Rescue

Main idea: Manifolds are spaces that *locally* "look like" $\mathbb{R}^n$, but may have a very different *global* "shape".

These are *ubiquitous* in robotics: orientations, rigid motions, similarity transformations, camera calibrations, viewing directions (for cameras) …



Why study manifolds?

- **It is proper:** These are the *mathematically correct* models to describe the things we care about

- **It is general:** Manifold theory provides a *unified* language for modeling *many different* spaces ⇒ "Only solve it once"

- **It gives you superpowers!** Employing proper mathematical models leads to algorithms that are *both faster and more accurate*

A few recent examples

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[1]

[1] Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[2] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[3] Department of Mathematics and Center for Data Science, Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA

IEEE TRANSACTIONS ON ROBOTICS, VOL. 33, NO. 1, FEBRUARY 2017 1

On-Manifold Preintegration for Real-Time Visual–Inertial Odometry

Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza

***Abstract*—Current approaches for visual–inertial odometry (VIO) are able to attain highly accurate state estimation via nonlinear optimization. However, real-time optimization quickly becomes infeasible as the trajectory grows over time; this problem is further emphasized by the fact that inertial measurements come at high rate, hence, leading to the fast growth of the number of variables in the optimization. In this paper, we address this issue by preintegrating inertial measurements between selected keyframes into single relative motion constraints. Our first contribution is a *preintegration theory* that properly addresses the manifold structure of the rotation group. We formally discuss the generative measurement model as well as the nature of the rotation noise and derive the expression for the *maximum a posteriori* state estimator. Our theoretical development enables the computation of all necessary Jacobians for the optimization and *a posteriori* bias correction in analytic form. The second contribution is to show that the preintegrated inertial measurement unit model can be seam-**

I. INTRODUCTION

THE use of cameras and inertial sensors for a three-dimensional (3-D) structure and motion estimation has received considerable attention from the robotics community. Both sensor types are cheap, ubiquitous, and complementary. A single moving camera is an exteroceptive sensor that allows us to measure appearance and geometry of a 3-D scene, up to an unknown metric scale; an inertial measurement unit (IMU) is a proprioceptive sensor that renders metric scale of monocular vision and gravity observable [1] and provides robust and accurate interframe motion estimates. Applications of a visual–inertial odometry (VIO) range from the autonomous navigation in GPS-denied environments, to the 3-D reconstruction, and augmented reality

Robotics: Science and Systems 2020
Corvalis, Oregon, USA, July 12-16, 2020

A Smooth Representation of Belief over SO(3) for Deep Rotation Learning with Uncertainty

Valentin Peretroukhin,[1,3] Matthew Giamou,[1] David M. Rosen,[2] W. Nicholas Greene,[3] Nicholas Roy,[3] and Jonathan Kelly[1]
[1]Institute for Aerospace Studies, University of Toronto;
[2]Laboratory for Information and Decision Systems,
[3]Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology

***Abstract*—Accurate rotation estimation is at the heart of robot perception tasks such as visual odometry and object pose estimation. Deep neural networks have provided a new way to perform these tasks, and the choice of rotation representation is an important part of network design. In this work, we present a novel symmetric matrix representation of the 3D rotation group, SO(3), with two important properties that make it particularly suitable for learned models: (1) it satisfies a smoothness property that improves convergence and generalization when regressing large rotation targets, and (2) it encodes a symmetric Bingham belief over the space of unit quaternions, permitting the training of uncertainty-aware models. We empirically validate the benefits of our formulation by training deep neural rotation regressors on two data modalities. First, we use synthetic point-cloud data to show that our representation leads to superior predictive accuracy over existing representations for arbitrary rotation targets. Second, we use image data collected onboard ground and aerial vehicles to demonstrate that our representation is amenable to an effective out-of-distribution (OOD) rejection technique that significantly improves the robustness of rotation estimates to unseen environmental effects and corrupted input**

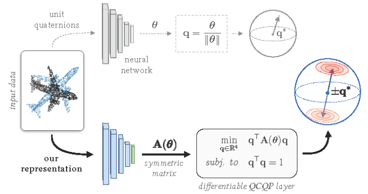


Fig. 1: We represent rotations through a symmetric matrix, $\mathbf{A}$, that defines a Bingham distribution over unit quaternions. To apply this representation to deep rotation regression, we present a differentiable layer parameterized by $\mathbf{A}$ and show how we can extract a notion of uncertainty from the spectrum of $\mathbf{A}$.

Best Paper Award (WAFR 2016)

IEEE Transactions on Robotics King-Sun Fu Best Paper Award (2018)

Best Student Paper Award (RSS 2020)

Application: Robotic mapping

Application: Camera tracking

Application: Rotation estimation

Plan of the day

- Manifolds
- A (very) brief introduction to groups
- An introduction to Lie groups

Disclaimer

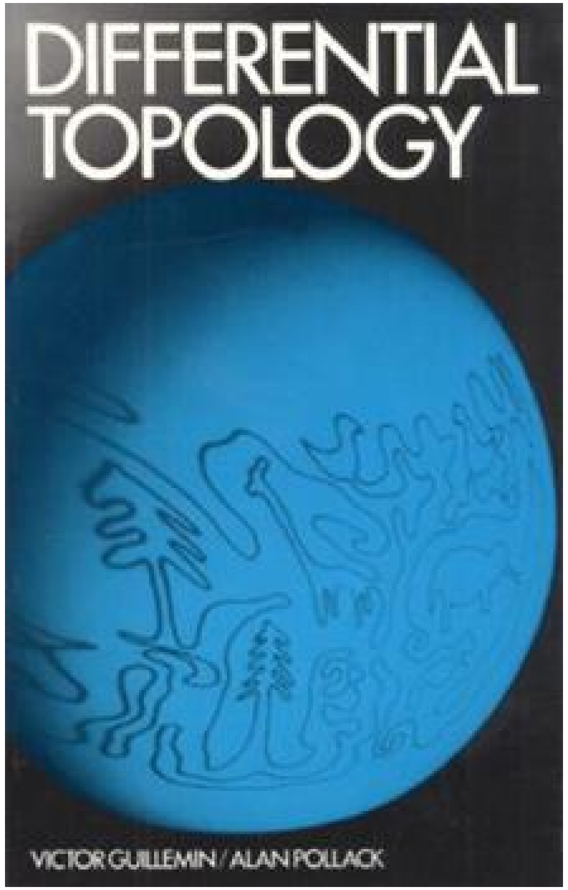
Manifolds, groups, and Lie groups can each fill a semester-long course

You will not be experts by the end of today (sorry :-P)

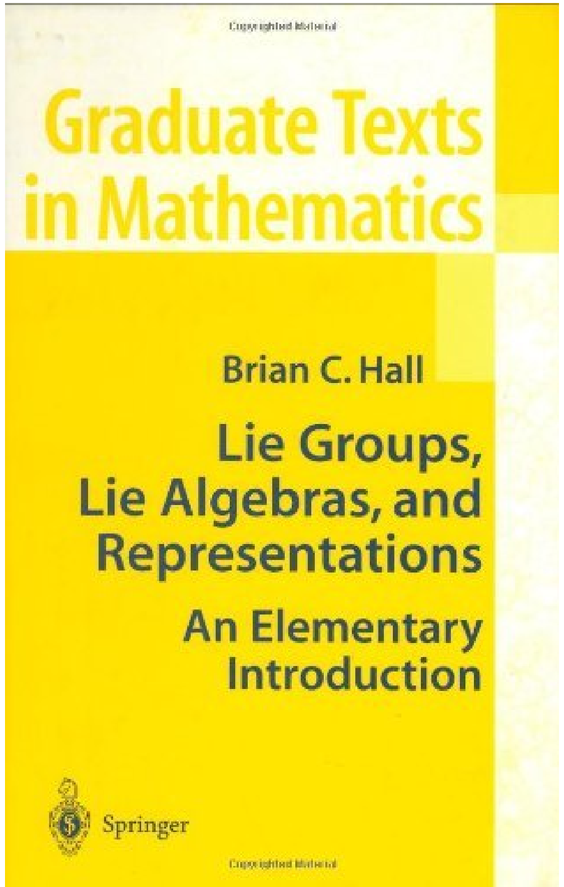
The goal of today’s lecture is to provide a high-level overview of:

- What these objects are
- Some of their properties that you should be aware of
- Why they are useful for robotics

References



Differential Topology (Guillemin and Pollack)



Lie Groups, Lie Algebras, and Representations (Hall)

Papers:

- J. Sola, J. Deray, D. Atchuthan: “A Micro Lie Theory for State Estimation in Robotics”
- E. Eade: “Lie Groups for 2D and 3D Transformations”

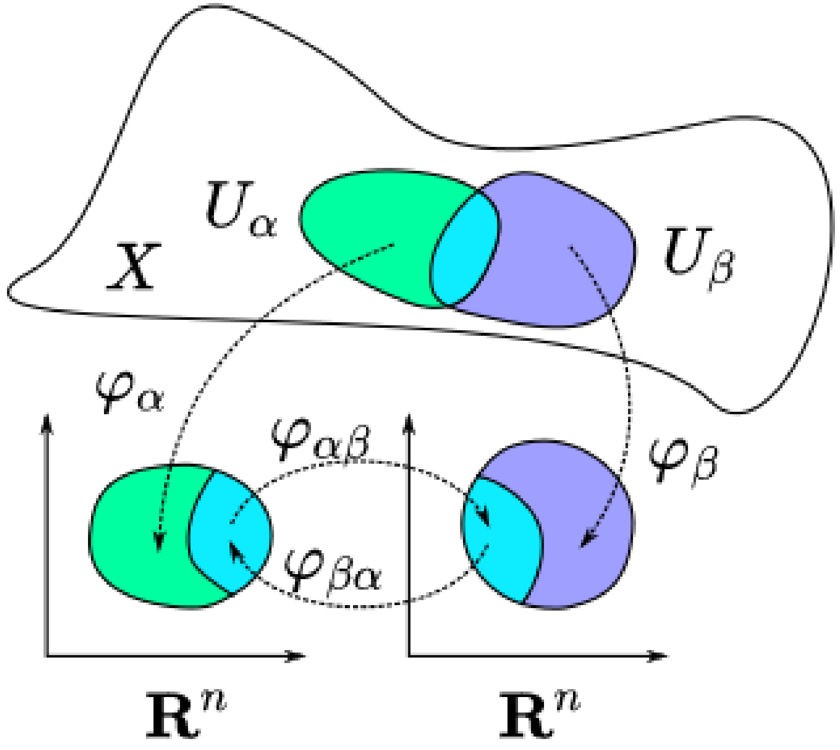
Manifold: Definition

A manifold X is a topological space in which each point $x \in X$ has an open set U and a continuous map $\varphi: U \to V \subseteq \mathbb{R}^n$ with a continuous inverse $\varphi^{-1}: V \to U$.

- The map φ is called a (*coordinate*) *chart*
  ⇒This *identifies* a subset of X with a subset of $\mathbb{R}^n$
  ⇒It lets us (*locally*!) define *coordinates* on X
  ⇒n is called the *dimension* of X

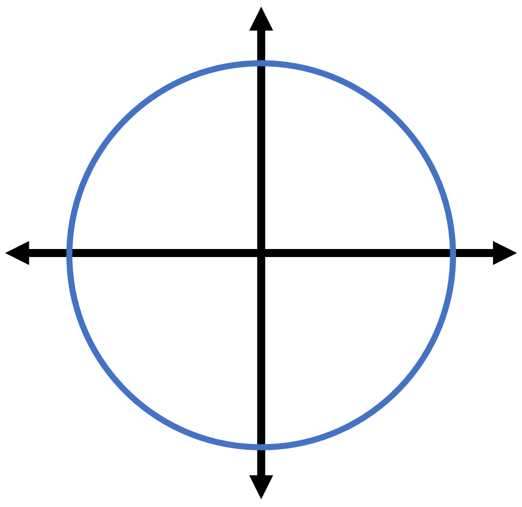
- The set of all *charts* on X is called an *atlas.*

- X is called a *smooth manifold* if additionally the *transition maps* $\varphi_{\alpha\beta} \triangleq \varphi_\beta \circ \varphi_\alpha^{-1}$ are differentiable (in $\mathbb{R}^n$) wherever they are defined
  ⇒ This lets us define *differentiation* on X (via $\mathbb{R}^n$)
  ⇒ We can do *calculus* on X



Example: The Unit Circle

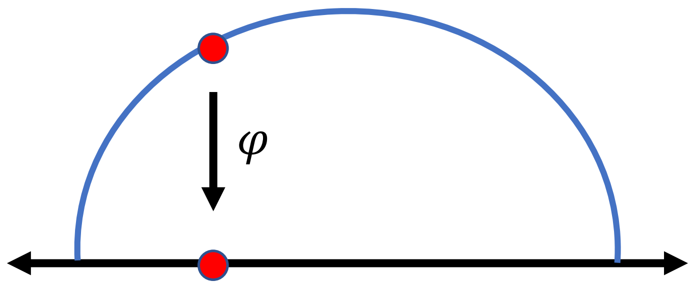
We can show that the circle is a smooth manifold by constructing a set of *charts* that cover it.

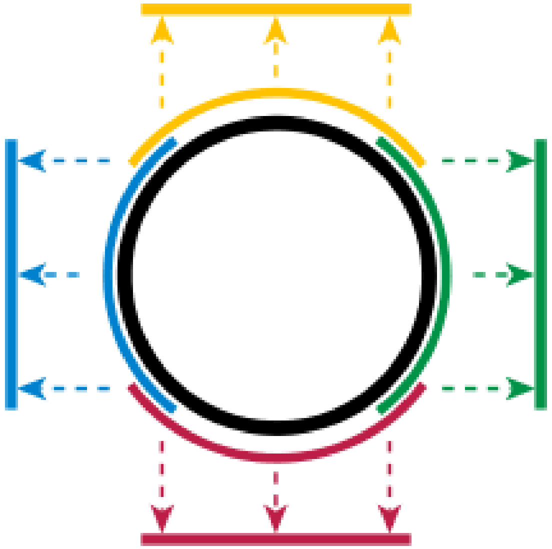


Q: What are some examples of these charts?

Ex: Project the upper semicircle onto the *x*-axis:

$$\varphi(x, y) = x.$$





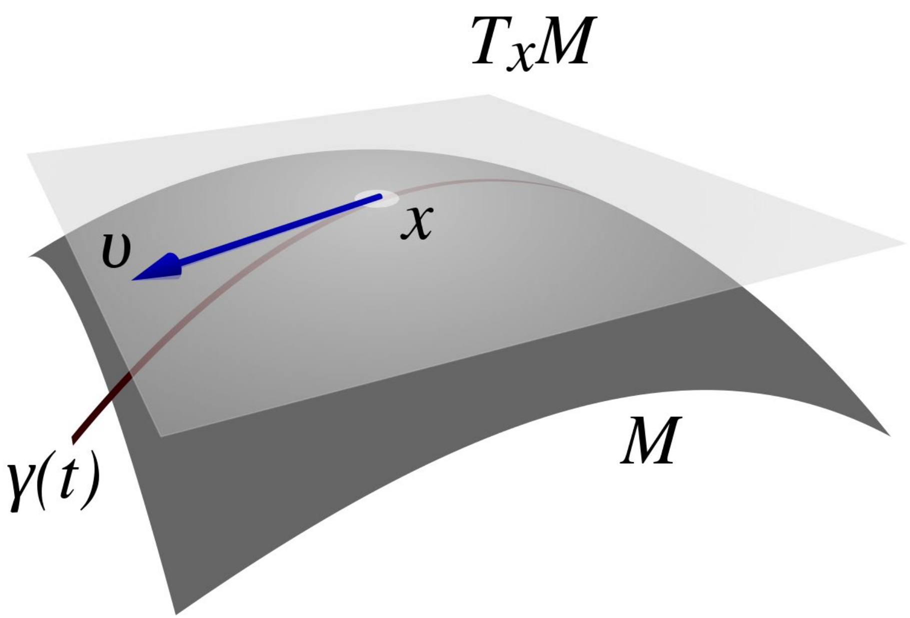
NB: In general we need *multiple* charts to cover an *entire* manifold

Tangent vectors and tangent spaces

Let *M* be a smooth manifold and x a point of *M*. A *tangent vector v* at x is:

- **Abstractly:** A *directional derivative* of smooth functions $f: M \to \mathbb{R}$.
- **Intuitively:** A (first-order) "*velocity*" along *M* at *x*.

The *tangent space* $T_x(M)$ is the set of **all** tangent vectors at *x*.



Key facts:

- The tangent space is a *linear space* of dimension dim(*M*).
- Intuitively, provides a *linear approximation* of *M* near *x*.
- For *submanifolds* $M \subseteq \mathbb{R}^n$ of Euclidean space, $T_x(M)$ is literally just the plane tangent to *M* at *x* (in the usual sense).

Calculus on manifolds

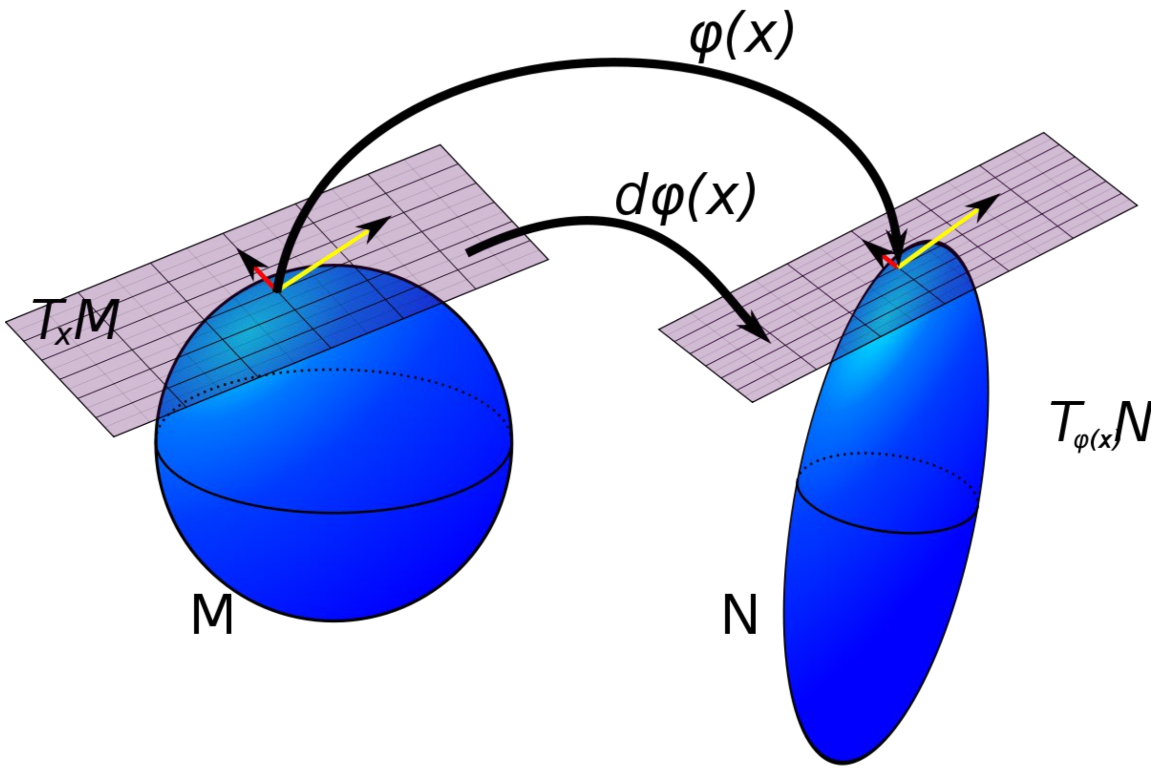
Let *M* and *N* be smooth manifolds, and $\varphi: M \to N$ a smooth map. At each *x* in *M*, the map φ induces a map $d\varphi_x: T_x(M) \longrightarrow T_{\varphi(x)}(N)$ between the tangent spaces of *M* and *N*, called the *derivative* of φ.

Intuitively: $d\varphi_x$ describes the infinitesimal change in $\varphi(x)$ under infinitesimal changes in *x*.

Key facts:

- The derivative $d\varphi_x: T_x(M) \longrightarrow T_{\varphi(x)}(N)$ is a *linear map* between the tangent spaces.
- If $M \subseteq \mathbb{R}^m$ and $N \subseteq \mathbb{R}^n$ are *submanifolds* of Euclidean spaces, then the derivative $d\varphi_x$ is just the usual Jacobian $\frac{\partial \varphi}{\partial x}$.

(Fun math fact: Derivatives are actually linear maps ☺!)



Diffeomorphisms

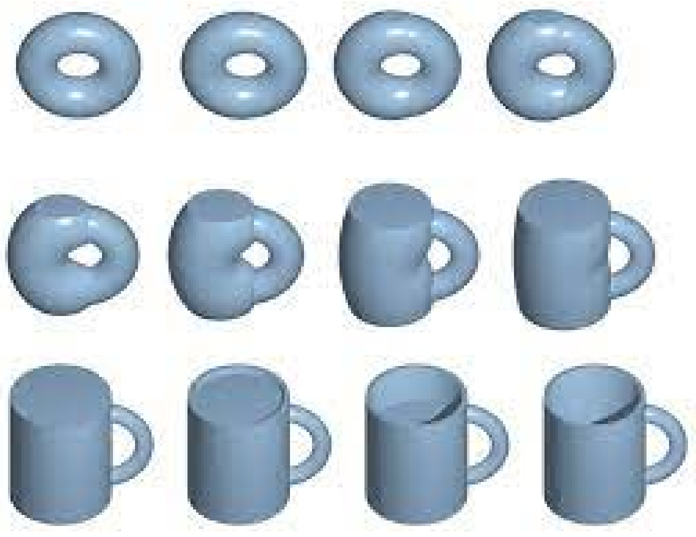
A smooth map $f: X \to Y$ is called a *diffeomorphism* if it has a smooth inverse $f^{-1}: Y \to X$.

Intuition: *X* and *Y* are *identical* as smooth manifolds

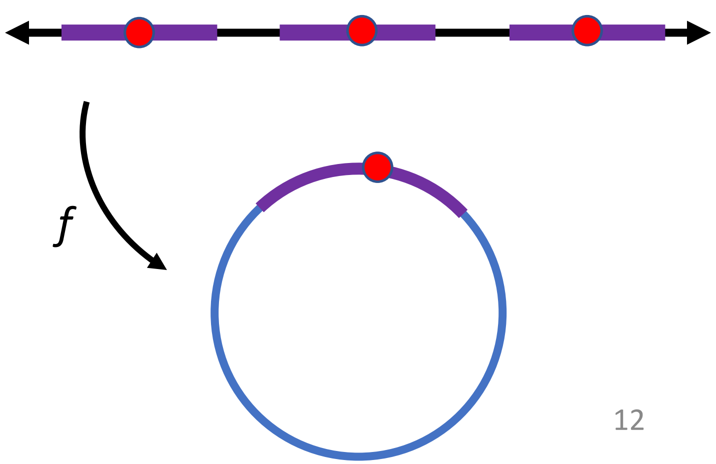
f is called a *local diffeomorphism* at $x \in X$ if there is an open set *U* around *x* such $f|_U: U \to f(U)$ is a diffeomorphism onto its image.

Intuition: *f* is *locally invertible* on *U.*

Example: $f(x) \equiv x \bmod 2\pi$



Visualization of a diffeomorphism between a coffee cup and a donut

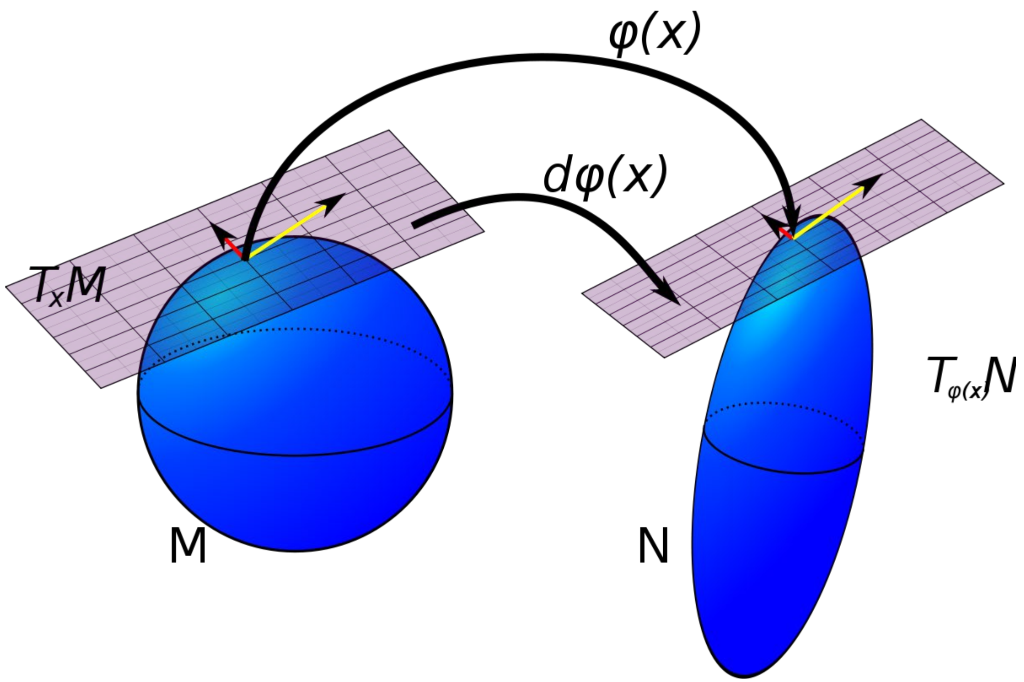


The Inverse Function Theorem

Theorem: Let $\varphi: M \to N$ be a smooth map, and suppose that $d\varphi_x: T_x(M) \to T_{\varphi(x)}(N)$ is an *isomorphism* at $x \in M$. Then f is a *local diffeomorphism* at x.

Key points

- This is *super powerful*: we can control φ in a *neighborhood* of x just by checking whether the derivative $d\varphi_x$ is invertible at a *point*!
- Super useful tool for proving stuff ☺!
- **NB:** In order for $d\varphi_x$ to be invertible, we must have dim(M) = dim(N)



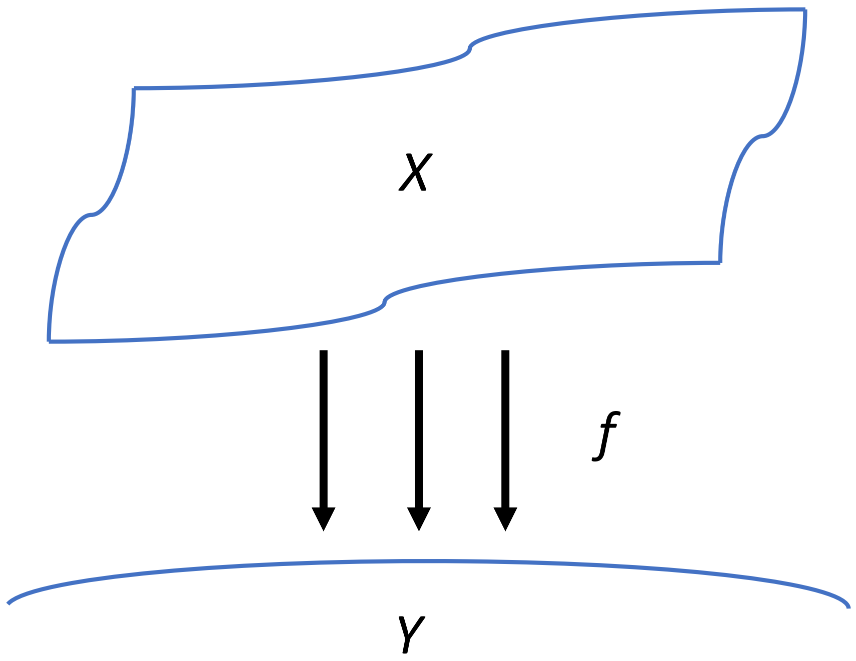
Submersions

A smooth map *f*: $X \to Y$ is called a (*local*) *submersion at* $x \in X$ if its derivative $df_x : T_x(X) \to T_{f(x)}(Y)$ is *surjective*.

f is called a *submersion* (unqualified) if df_x is a surjective for **all** $x \in X$.

Local Submersion Theorem: Suppose that *f*: $X \to Y$ is a local submersion at $x \in X$. Then there exist coordinate charts around *x* and *y* such that:

$$f(x_1, \ldots, x_k) = (x_1, \ldots, x_l)$$



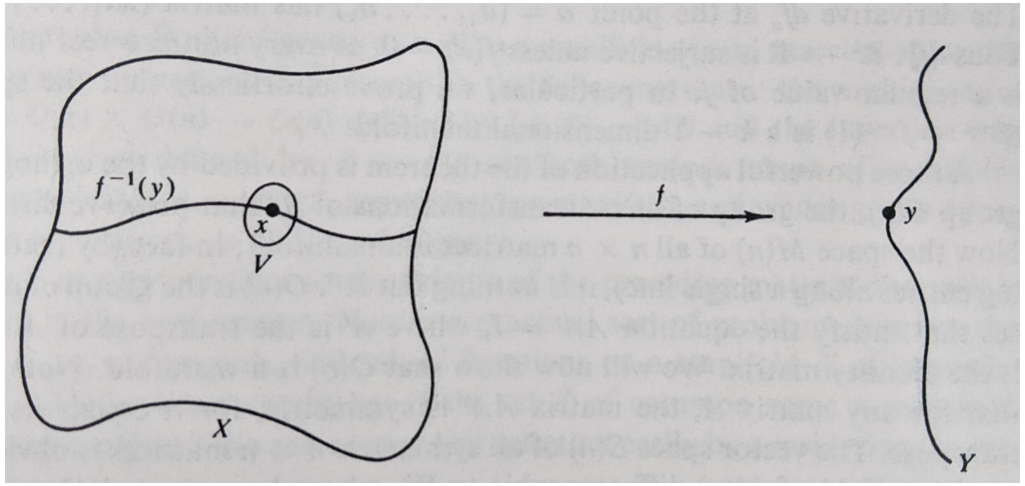
Intuition: *f* locally "looks like" a projection!

Preimage Theorem

Let $f: X \to Y$ be a smooth map. A point $y \in Y$ is called a *regular value* if f is a submersion at *all* of y's preimages $f^{-1}(y)$.

Preimage Theorem: Let y be a regular value of f. Then y's preimage $P \triangleq f^{-1}(y)$ is an *embedded submanifold* of X of dimension $\dim(X) - \dim(Y)$, and

$$T_x(P) = \ker df_x \quad \text{for all } x \in P.$$



Key point: This is *super powerful* -- we can show that *sets* are actually *smooth manifolds* by choosing the right f, X and Y.

- Super useful tool for proving stuff ☺!
- No coordinate charts!
- We also get a description of P's tangent spaces!

Example: Orthogonal Group

Let's show that the orthogonal group $O(n) \coloneqq \{R \in \mathbb{R}^{n\times n} | R^T R = I\}$ is a smooth manifold.

By definition, we have $O(n) \triangleq f^{-1}(I)$ for $f\colon \mathbb{R}^{n\times n} \to Sym(n), f(R) \triangleq R^T R$.

So we need to show that *I* is a *regular value*.

The derivative of f at *R* is:

$$df_R[\dot{R}] = R^T\dot{R} + \dot{R}^T R = 2Sym\left(R^T\dot{R}\right).$$

Since the symmetrization map $Sym\colon \mathbb{R}^{n\times n} \to Sym(n)$ is surjective and R^T is invertible for *R* ∈ O(n), df_R is *surjective* for all *R* ∈ O(n)

⇒ *I* is a *regular value* for *f*.

⇒O(*n*) is a smooth manifold (by Preimage Theorem).

Example: Orthogonal Group

Other stuff we get (basically) for free:

Dimension: $\dim(\mathbb{R}^{n\times n}) = n^2$ and $\dim\big(Sym(n)\big) = n(n+1)/2$, so

$$\dim\big(O(n)\big) = n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}.$$

Tangent spaces: Recall $T_R(O(n)) = \ker(df_R)$. Since $df_R[\dot{R}] = R^T\dot{R} + \dot{R}^T R$,

$$\ker(df_R) = \left\{\dot{R} \in \mathbb{R}^{n\times n} : R^T\dot{R} + \dot{R}^T R = 0\right\}.$$

Special case: At *R* = *I*:

$$T_I(O(n)) = \left\{\dot{R} \in \mathbb{R}^{n\times n} : \dot{R} + \dot{R}^T = 0\right\} = Skew(n)$$

Exercise: The Sphere S^n

The sphere S^n is the set of unit vectors in (n+1)-dimensional space:

$$S^n = \{x \in \mathbb{R}^{n+1} \mid \|x\| = 1\}.$$

- Prove that S^n is a smooth manifold, and calculate its dimension

- What are its tangent spaces?

Solution

Let's use $f\colon \mathbb{R}^{n+1} \to \mathbb{R}$, $f(x) \triangleq \|x\|^2$. Then $S^n = f^{-1}(1)$.

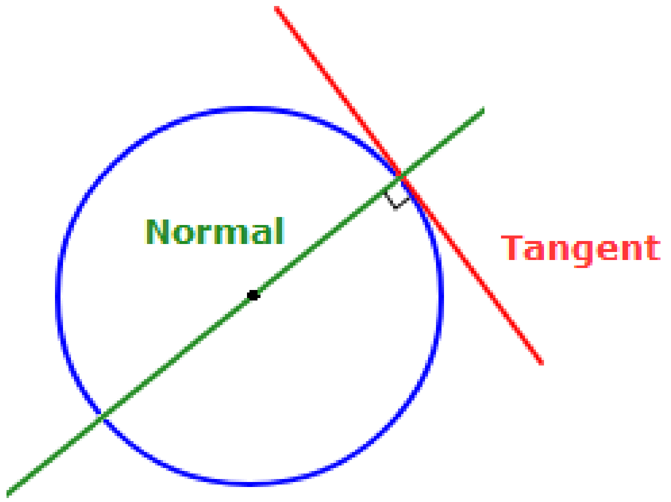
We need to show that 1 is a *regular value*.

The derivative of f at $x \in S^n$ is $df_x[\dot{x}] = 2x^T\dot{x}$.

Since $x \neq 0$ for all $x \in S^n$, then given any $c \in \mathbb{R}$, we can find some tangent vector $\dot{x} \in \mathbb{R}^{n+1}$ such that $c = 2x^T\dot{x}$, so df_x is *surjective* for all $x \in S^n$.

$\Rightarrow$ 1 is a *regular value* for *f*.

$\Rightarrow$ S^n is a smooth manifold (by Preimage Theorem).



Dimension: $\dim(S^n) = \dim(\mathbb{R}^{n+1}) - \dim(\mathbb{R}) = n$

Tangent spaces: $T_x(S^n) = \ker(df_x) = \{\dot{x} \in \mathbb{R}^{n+1} \mid x^T\dot{x} = 0\}$
$\Rightarrow$ $T_x(S^n)$ is the set of vectors orthogonal to x.

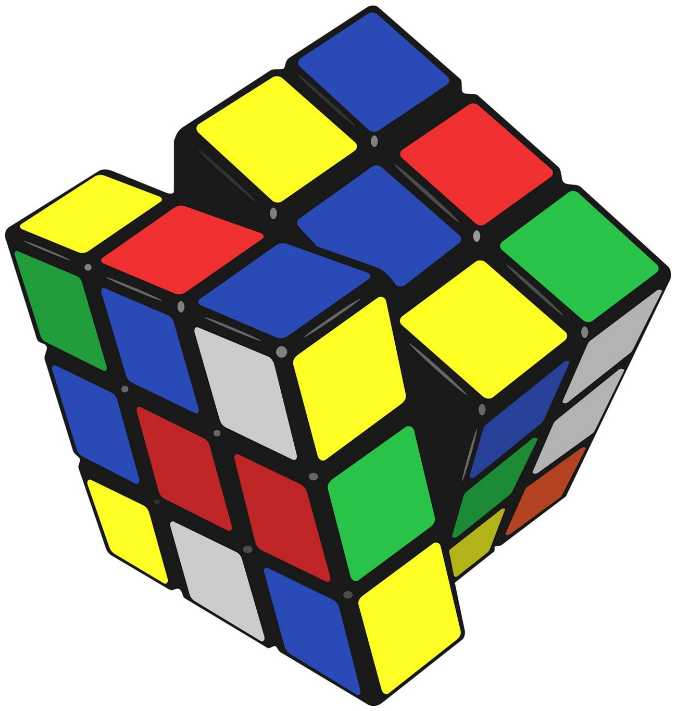
Plan of the day

- Manifolds
- A (very) brief introduction to groups
- An introduction to Lie groups

Groups

A **group** is a set G together with a binary operation $\cdot$ that satisfies the following three *group axioms*:

- **Associativity:** $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ for all $x, y, z \in G$
- **Identity:** There is an *identity element* $e \in G$ such that $e \cdot g = g \cdot e = g$ for all $g \in G$.
- **Invertibility:** Every element $g \in G$ has an *inverse* in G, denoted g^{-1}, satisfying $gg^{-1} = g^{-1}g = e$.



Group theory can help you solve the Rubik's Cube *suuuuuper fast!*

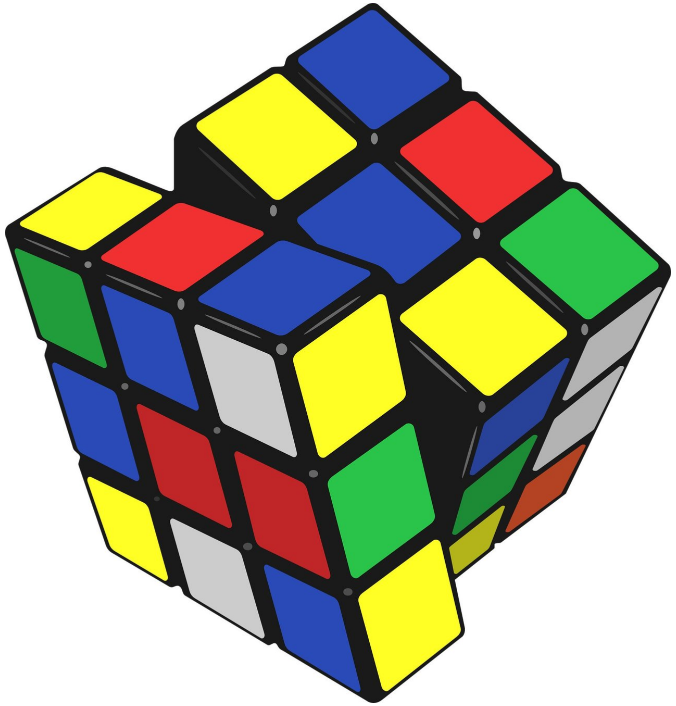
Key points:

- Super simple to define, but an *extremely rich* theory
- **LOTS** of cool examples ☺!
  - Integers under addition
  - Symmetry / transformation groups for objects
  - **Matrix groups** ☺!

Group actions

Let G be a group with identity e and X a set. A *left-group action* of G on X is a binary map $\star: G \times X \to X$ satisfying the following two axioms:

- **Identity:** $e \star x = x$ for all $x \in X$.
- **Compatibility:** $g \star (h \star x) = gh \star x$ for all $g, h \in G$ and $x \in X$.



Key points:

- Group actions are one of the most important applications of group theory
(Historical note: Group theory was originally developed to study symmetry groups of objects!

- **Import examples:** Matrix group actions!
  - SO(n) acts on $\mathbb{R}^n$ by *rotation*.
  - GL(n) acts on $\mathbb{R}^n$ by *linear* transforms
  - Aff(n) acts on $\mathbb{R}^n$ by *affine transforms* (coordinate frame transformations).

Plan of the day

- Manifolds
- A (very) brief introduction to groups
- An introduction to Lie groups

Lie Groups

A *Lie group* G is a *group* that is *also* a *smooth manifold*, in which the group multiplication and inversion maps are smooth.

In this case, we say that the group and manifold structures are “compatible”.

Key points:

- This is an *extremely strong* condition
  ⇒ Lie groups are “highly regular” objects (more on this in a moment …)
  ⇒ **LOTS** of things we can say about them that don’t hold for general manifolds

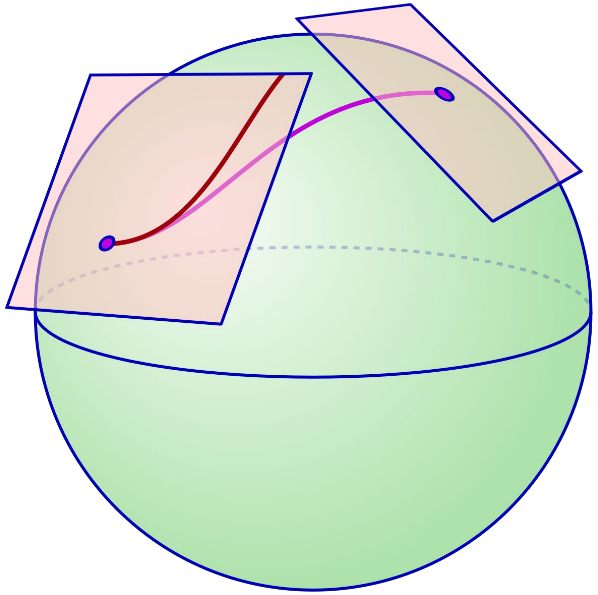
- They are super pretty! 😍

Motivating example: Matrix Lie groups!

A first example: The left-multiplication map

To each $g \in G$ we associate the following smooth *left-multiplication map*:

$$L_g : G \to G$$
$$L_g(x) = gx.$$



Key points:

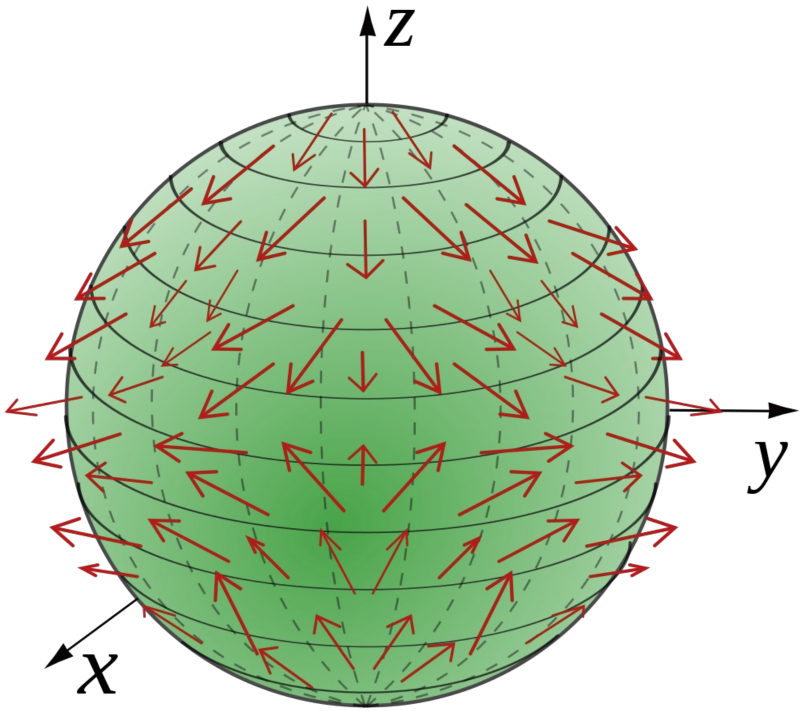
- L_g is a smooth map, and it has a smooth inverse $L_{g^{-1}}$
  $\Rightarrow$ L_g is a *diffeomorphism* for all $g \in G$

- Given any two points $x, y \in G$, $L_{yx^{-1}}(x) = y$
  $\Rightarrow$ There is a symmetry of G sending *any* point x onto *any* other point y

  $\Rightarrow$ G is a *homogeneous space*: all points $g \in G$ "look the same"

- Since L_x is a diffeomorphism, its derivative $dL_x : T_e(G) \to T_x(G)$ is an isomorphism for all x.
  $\Rightarrow$ We have a *canonical* way of identifying *every* tangent space $T_x(G)$ with the *single* "standard" tangent space $T_e(G)$ at the identity

The Lie Algebra and Left-Invariant Vector Fields



A *vector field* on a smooth manifold M is a smooth map V that assigns to each x in M a tangent vector $V(x) \in T_x(M)$.

A vector field V on a Lie group G is called *left-invariant* if:

$$dL_g V(x) = V(gx).$$

Key point: A left-invariant vector field is *completely determined* by its value at the identity element $e \in G$.

This gives an *identification*: $T_e(G) \Leftrightarrow$ {left-invariant vector fields on G}:

$$\omega \Leftrightarrow V_\omega$$

where V_ω is the left-invariant vector field defined by:

$$V_\omega(x) \triangleq dL_x(\omega)$$

Key point: We can identify a *velocity* $v \in T_x(M)$ at an *arbitrary* point $x \in G$ with a vector ω in the *fixed* tangent space $T_e(G)$.

We call the tangent space $T_e(G)$ the ***Lie algebra*** of G. We can think of this as the set of *infinitesimal generators for motion* on G.

The Exponential Map

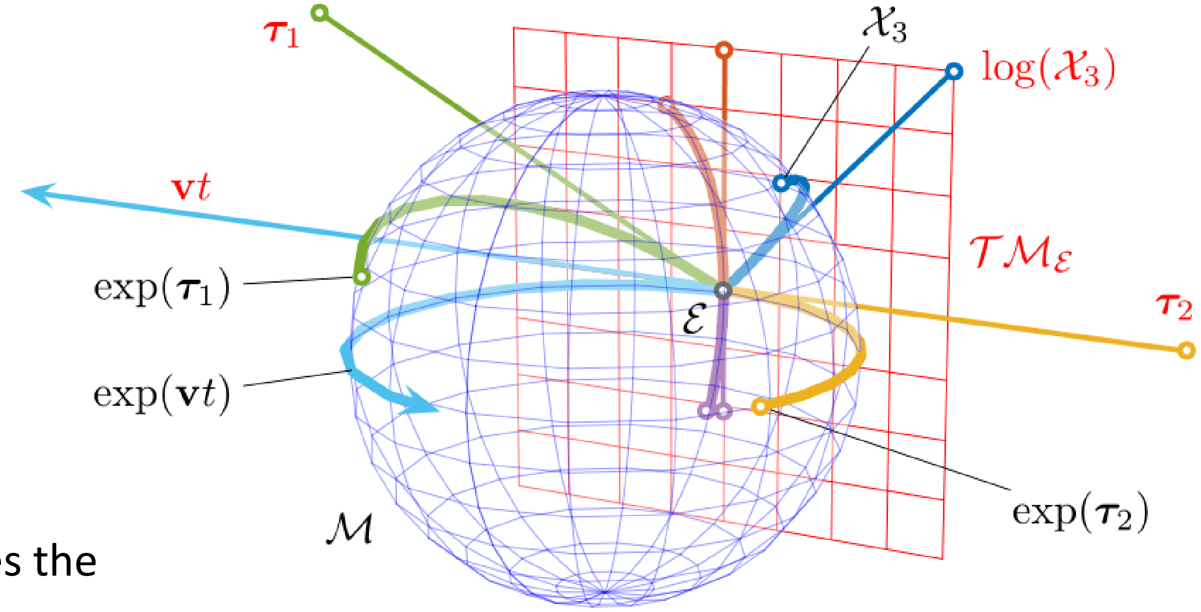
Recap: We just saw that every vector ω in $T_e(G)$ generates a left-invariant vector field on V_ω.

Key question: What is the *flow* on G generated by V_ω?

The *exponential map* provides the answer. This is a map

$$exp: T_e(G) \to G$$

from the Lie *algebra* into the Lie *group.*



Key property: The unique *integral curve* $c(t)$ of V_ω that takes the value x in G at time t = 0 is:

$$c(t) = x \exp(t\omega)$$

⇒ This tells us how to *move* on G along the flow of ω.

Special case: for *matrix groups* (subgroups of GL(n)), the exponential map has an especially nice form:

$$\exp(X) = \sum_{k=0}^{\infty} \frac{X^k}{k!}$$

Lie Groups: Putting it all together

Main things to take away:

- Many objects of interest in robotics naturally belong to *non-Euclidean manifolds*, specifically *Lie groups*.
  Ex: Rotations, rigid motions, coordinate transforms, etc.

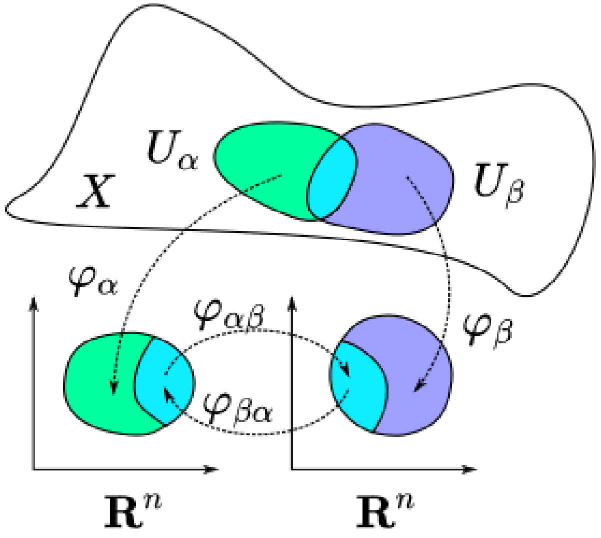
- Because manifolds are *locally* Euclidean, we can still (locally) stick coordinates on them and do calculus (yay ☺!)

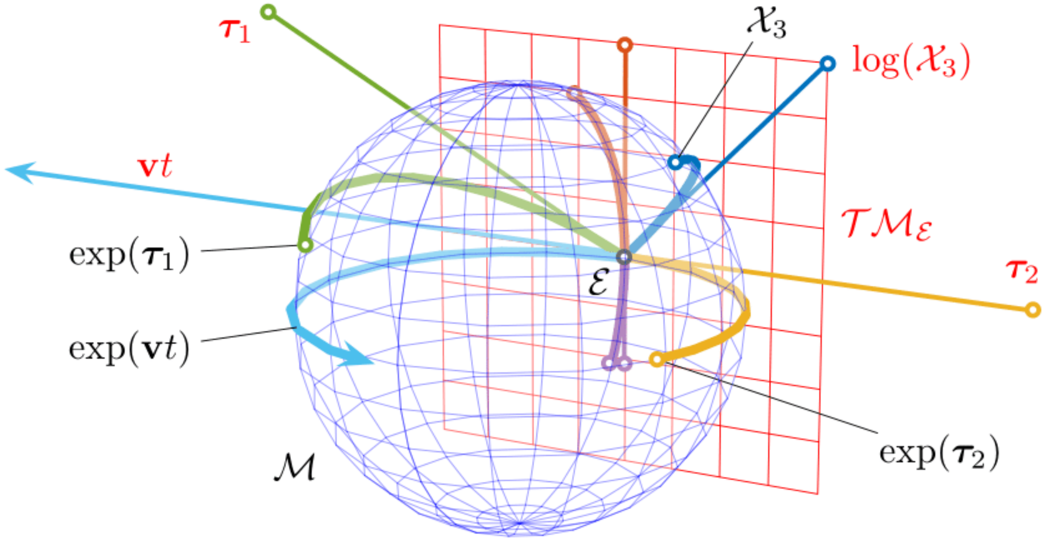
Lie groups are *especially* nice because:

- We can describe *velocities* *any* point using an element of the *Lie algebra* $T_e(G)$. This is *always* a *fixed* linear space (isomorphic to $\mathbb{R}^n$), no matter how complicated G is.

- We can *move* along a tangent vector ω using the group's *exponential map*. This mapping *automatically* respects the group's geometry (i.e. *satisfies constraints*).

Punchline: Lie group theory provides a *principled*, *unified language* for building algorithms on *arbitrary* Lie groups!





A Non-Exhaustive List of Commonly-Used Lie Groups in Robotics and Computer Vision

Lie group $\mathcal{M}, \circ$		size	dim	$\mathcal{X} \in \mathcal{M}$	Constraint	$\boldsymbol{\tau}^\wedge \in \mathfrak{m}$	$\boldsymbol{\tau} \in \mathbb{R}^m$	$\mathrm{Exp}(\boldsymbol{\tau})$	Comp.	Action
n-D vector	$\mathbb{R}^n, +$	n	n	$\mathbf{v} \in \mathbb{R}^n$	$\mathbf{v} - \mathbf{v} = \mathbf{0}$	$\mathbf{v} \in \mathbb{R}^n$	$\mathbf{v} \in \mathbb{R}^n$	$\mathbf{v} = \exp(\mathbf{v})$	$\mathbf{v}_1 + \mathbf{v}_2$	$\mathbf{v} + \mathbf{x}$
circle	$S^1, \cdot$	2	1	$\mathbf{z} \in \mathbb{C}$	$\mathbf{z}^*\mathbf{z} = 1$	$i\theta \in i\mathbb{R}$	$\theta \in \mathbb{R}$	$\mathbf{z} = \exp(i\theta)$	$\mathbf{z}_1\,\mathbf{z}_2$	$\mathbf{z}\,\mathbf{x}$
Rotation	$SO(2), \cdot$	4	1	$\mathbf{R}$	$\mathbf{R}^\top\mathbf{R} = \mathbf{I}$	$[\theta]_\times \in \mathfrak{so}(2)$	$\theta \in \mathbb{R}$	$\mathbf{R} = \exp([\theta]_\times)$	$\mathbf{R}_1\,\mathbf{R}_2$	$\mathbf{R}\,\mathbf{x}$
Rigid motion	$SE(2), \cdot$	9	3	$\mathbf{M} = \left[\begin{smallmatrix}\mathbf{R} & \mathbf{t}\\ 0 & 1\end{smallmatrix}\right]$	$\mathbf{R}^\top\mathbf{R} = \mathbf{I}$	$\left[\begin{smallmatrix}[\theta]_\times & \boldsymbol{\rho}\\ 0 & 0\end{smallmatrix}\right] \in \mathfrak{se}(2)$	$\left[\begin{smallmatrix}\boldsymbol{\rho}\\ \theta\end{smallmatrix}\right] \in \mathbb{R}^3$	$\exp\left(\left[\begin{smallmatrix}[\theta]_\times & \boldsymbol{\rho}\\ 0 & 0\end{smallmatrix}\right]\right)$	$\mathbf{M}_1\,\mathbf{M}_2$	$\mathbf{R}\,\mathbf{x} + \mathbf{t}$
3-sphere	$S^3, \cdot$	4	3	$\mathbf{q} \in \mathbb{H}$	$\mathbf{q}^*\mathbf{q} = 1$	$\boldsymbol{\theta}/2 \in \mathbb{H}_p$	$\boldsymbol{\theta} \in \mathbb{R}^3$	$\mathbf{q} = \exp(\mathbf{u}\theta/2)$	$\mathbf{q}_1\,\mathbf{q}_2$	$\mathbf{q}\,\mathbf{x}\,\mathbf{q}^*$
Rotation	$SO(3), \cdot$	9	3	$\mathbf{R}$	$\mathbf{R}^\top\mathbf{R} = \mathbf{I}$	$[\boldsymbol{\theta}]_\times \in \mathfrak{so}(3)$	$\boldsymbol{\theta} \in \mathbb{R}^3$	$\mathbf{R} = \exp([\boldsymbol{\theta}]_\times)$	$\mathbf{R}_1\,\mathbf{R}_2$	$\mathbf{R}\,\mathbf{x}$
Rigid motion	$SE(3), \cdot$	16	6	$\mathbf{M} = \left[\begin{smallmatrix}\mathbf{R} & \mathbf{t}\\ 0 & 1\end{smallmatrix}\right]$	$\mathbf{R}^\top\mathbf{R} = \mathbf{I}$	$\left[\begin{smallmatrix}[\boldsymbol{\theta}]_\times & \boldsymbol{\rho}\\ 0 & 0\end{smallmatrix}\right] \in \mathfrak{se}(3)$	$\left[\begin{smallmatrix}\boldsymbol{\rho}\\ \boldsymbol{\theta}\end{smallmatrix}\right] \in \mathbb{R}^6$	$\exp\left(\left[\begin{smallmatrix}[\boldsymbol{\theta}]_\times & \boldsymbol{\rho}\\ 0 & 0\end{smallmatrix}\right]\right)$	$\mathbf{M}_1\,\mathbf{M}_2$	$\mathbf{R}\,\mathbf{x} + \mathbf{t}$

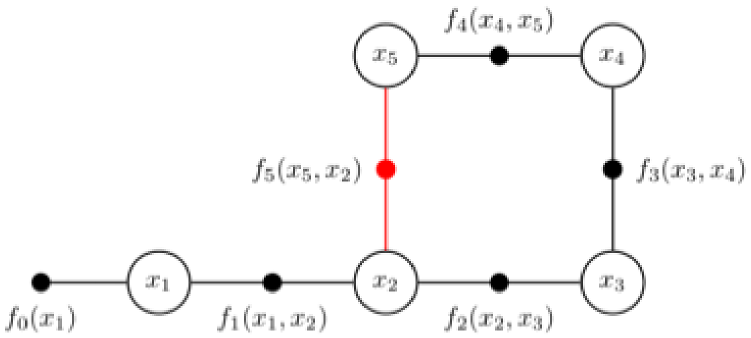
Example / Preview: The GTSAM Library

- C++ / Python / MATLAB library for optimizing functions over Lie groups

- Functions are specified using *factor graphs*: graphical models describing how to build complex objectives from simple summands

- Makes it easy to model and solve complex robotic perception and control problems!

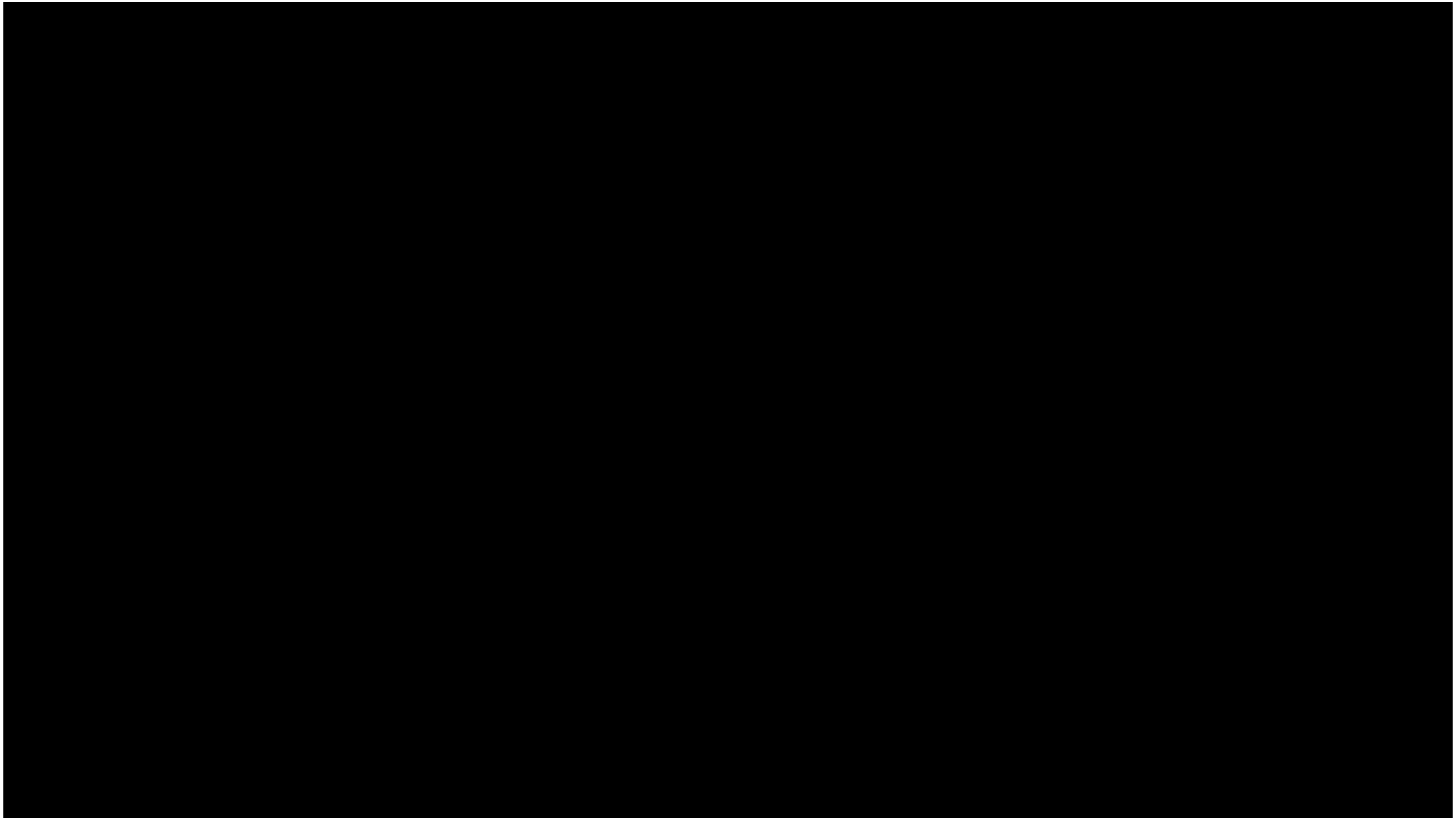
- **LOTS** of applications: Robotic mapping, visual-inertial odometry, 3D visual reconstruction, …



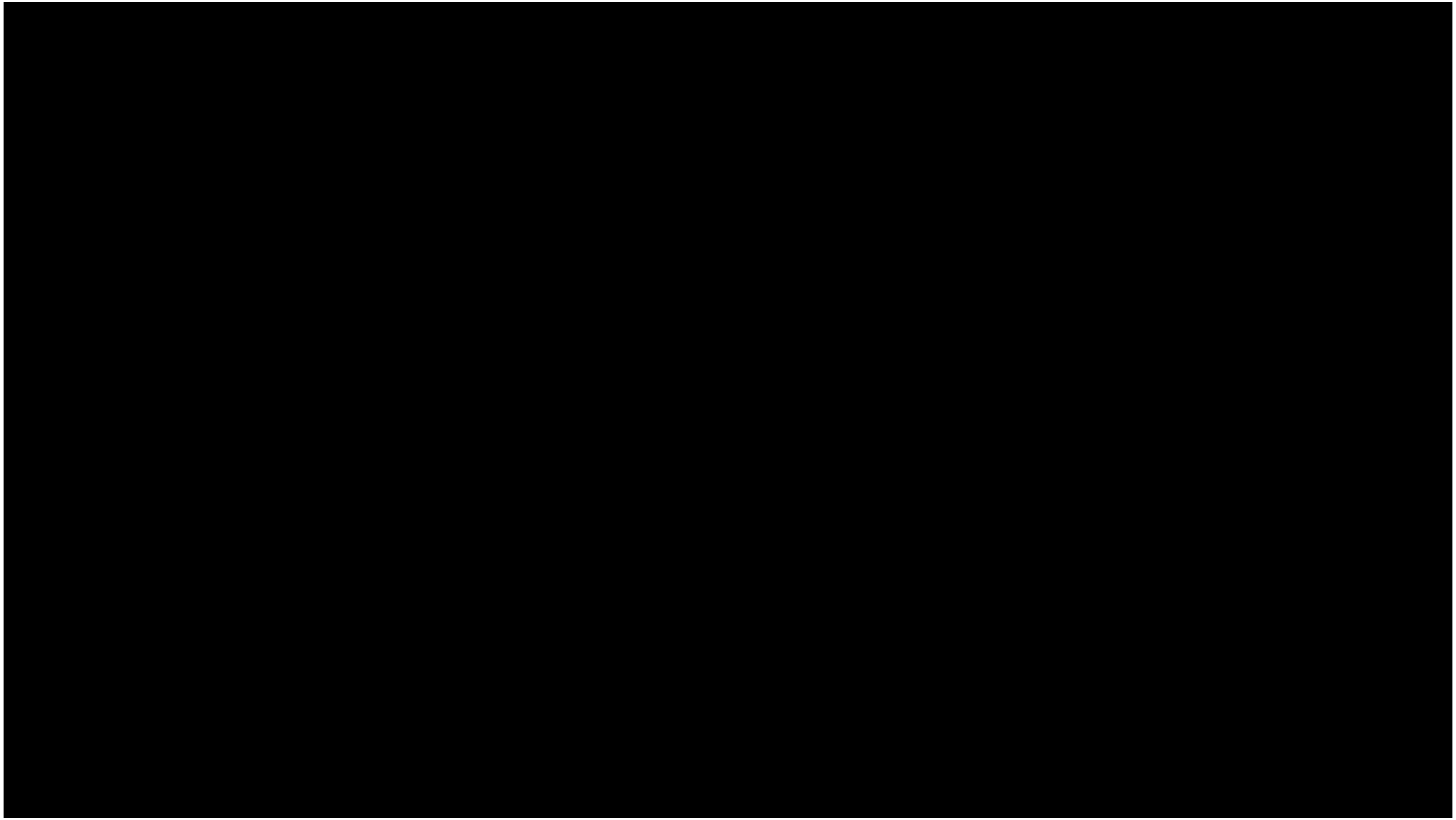
$$f(X) = \sum_{k \epsilon F} f_k(\Theta_k)$$

More on this later in the course …

Example: Visual-Inertial Odometry



Example: Learning to Estimate Rotations



Lie Groups: Putting it all together

Main things to take away:

- Many objects of interest in robotics naturally belong to *non-Euclidean manifolds*, specifically *Lie groups*.
  Ex: Rotations, rigid motions, coordinate transforms, etc.

- Because manifolds are *locally* Euclidean, we can still (locally) stick coordinates on them and do calculus (yay ☺!)

Lie groups are *especially* nice because:

- We can describe *velocities any* point using an element of the *Lie algebra* $T_e(G)$. This is *always* a *fixed* linear space (isomorphic to $\mathbb{R}^n$), no matter how complicated G is.

- We can *move* along a tangent vector ω using the group's *exponential map*. This mapping *automatically* respects the group's geometry (i.e. *satisfies constraints*).

Punchline: Lie group theory provides a *principled*, *unified language* for building algorithms on *arbitrary* Lie groups!