Differential Geometry in Mathlib

Michael B. Rothgang (he/him)

Formalised mathematics group
Universität Bonn

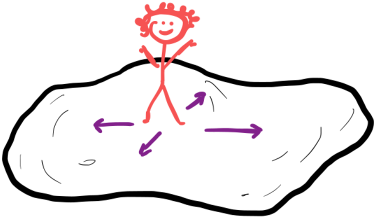
ItaLean workshop
December 11, 2025

Plan for today

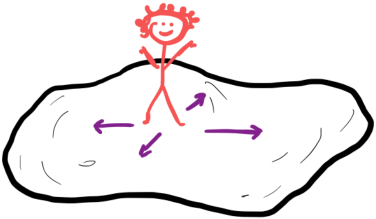
Differential geometry. . .

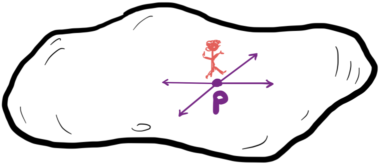
... is the study of smooth manifolds and their Lie group, Riemannian, symplectic, foliations, vector bundle, . . . structures

What is a manifold?



What is a manifold?

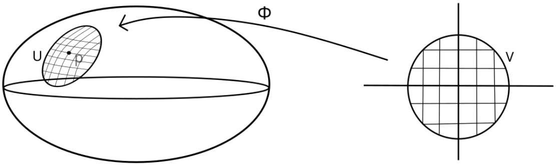




surface of a potato is a manifold: locally looks like a disk

Smooth manifolds

- topological **manifold**: second countable Hausdorff topological space M locally homeomorphic to open ball in $\mathbb{R}^n$
- every $p \in M$ has a coordinate chart: $p \in U \subset M$ open, homeomorphism $\phi \colon V \to U$ for $V \subset \mathbb{R}^n$ open ball
- **smooth manifold**: all coordinate transformations from overlapping charts are smooth



Picture courtesy of Dominik Gutwein.

Examples of smooth manifolds

- empty set (of any dimension)
- 0-dimensional: isolated points
- 1-dimensional: $\mathbb{R}$, $\mathbb{S}^1$
- n-dimensional: open disc $\mathbb{D} \subset \mathbb{R}^n$
- $n = 2$: $\mathbb{R}^2$, $\mathbb{S}^2$, $\mathbb{T}^2$, Σ_g for $g \geq 1$

Examples of smooth manifolds

- empty set (of any dimension)
- 0-dimensional: isolated points
- 1-dimensional: $\mathbb{R}$, $\mathbb{S}^1$
- n-dimensional: open disc $\mathbb{D} \subset \mathbb{R}^n$
- $n = 2$: $\mathbb{R}^2$, $\mathbb{S}^2$, $\mathbb{T}^2$, Σ_g for $g \geq 1$

- $n \geq 3$: complicated; classification for $n \geq 4$ impossible
- $\mathbb{R}^n$, $\mathbb{S}^n$, $\mathbb{T}^n$, $\mathbb{RP}^n$, $\mathbb{CP}^n$,
  $\{[z_0 : z_1 : z_2 : z_3 : z_4] \in \mathbb{CP}^4 \mid z_0^5 + \cdots + z_4^5 = 0\}$
  configuration spaces in physics and engineering
- **not** a manifold: letter "X"

Manifolds with boundary or corners

- $\overline{\mathbb{D}} \subset \mathbb{R}^n$ is a manifold with **boundary**
- interior points locally look like open ball in $\mathbb{R}^n$,
  boundary points look like open ball in upper half of $\mathbb{R}^n$
- manifolds with **corners**: local model is Euclidean quadrant
  $[0, 1]^n \subset \mathbb{R}^n$ has corners

Criteria for a good mathlib definition

- number of pre-requisites required
- ergonomic to work with

Criteria for a good mathlib definition

- number of pre-requisites required
- ergonomic to work with
- generality: sufficient for all modern research

Criteria for a good mathlib definition

- number of pre-requisites required
- ergonomic to work with
- generality: sufficient for all modern research
  $\Rightarrow$ abstraction to allow re-use

Constraints for defining manifolds

- allow different smoothness
- allow boundary and corners
- allow different base field: $\mathbb{R}$, $\mathbb{C}$ or p-adic numbers $\mathbb{Q}_p$
- infinite-dimensional manifolds (e.g. $C^k(M, N)$)

Formalising manifolds with boundary

A smooth manifold includes several data:

- M: the manifold (e.g. $\overline{\mathbb{D}}$)
- H: the local model, a topological space (e.g. $\mathbb{H}$)

Formalising manifolds with boundary

A smooth manifold includes several data:

- M: the manifold (e.g. $\overline{\mathbb{D}}$)
- H: the local model, a topological space (e.g. $\mathbb{H}$)
- E: normed space (e.g. $\mathbb{R}^2$)
- I: model with corners, continuous map $H \to E$ (e.g. canonical inclusion)

Formalising manifolds with boundary

A smooth manifold includes several data:

- M: the manifold (e.g. $\overline{\mathbb{D}}$)
- H: the local model, a topological space (e.g. $\mathbb{H}$)
- E: normed space (e.g. $\mathbb{R}^2$)
- I: model with corners, continuous map $H \to E$ (e.g. canonical inclusion)
- charts on M (one preferred chart at each point)
- compatibility condition: transition maps lie in structure groupoid

Questions so far?

If you're fine with the mathematics, let's look at the formalisation in Lean.

A tour of differential geometry in mathlib

Overview: differential geometry in mathlib

- smooth manifolds, smooth maps, (continuous) differentiability
- (manifold) Fréchet derivative, chain rule
- diffeomorphisms, local diffeomorphisms; smooth immersions
- products and disjoint unions of manifolds
- classification of 0-dimensional manifolds
- examples: $\mathbb{R}^n$, half-space, quadrants; intervals
  unit sphere; units in Lie groups

Overview: differential geometry in mathlib (cont.)

- (topological and smooth) vector bundles
- basic constructions: trivial bundle, direct sum, product bundle, hom bundle, (co)tangent bundle
- (continuous) differentiability of sections, smooth bundle maps
- Lie bracket of vector fields; Lie groups and their Lie algebra
- smooth bundle metrics; (basic) Riemannian manifolds
- existence of integral curves and local flows
- smoothness of local flows

What's missing?

- special maps: immersions, embeddings, submersions
- smooth submanifolds; sub-bundles
- quotients of manifolds; gluing
- implicit and inverse function theorems
- constant rank theorem; regular value theorem
- existence of a Riemannian metric
- differential topology: Sard's theorem
- classification of 1-manifolds and 2-manifolds
- smooth fibre bundles

In the making/in progress

- smooth immersions and embeddings, submanifolds
- covariant derivatives/connections
  very close: Ehresmann connections, pullback connection; geodesic flow, exponential map
- existence and uniqueness of the Levi-Civita connection
- Quotient manifolds

In the making/in progress

- smooth immersions and embeddings, submanifolds
- covariant derivatives/connections
  very close: Ehresmann connections, pullback connection; geodesic flow, exponential map
- existence and uniqueness of the Levi-Civita connection
- Quotient manifolds
- unoriented bordism groups
- oriented manifolds — almost done, help welcome
- inverse function theorem — help welcome
- Moreira's version of Morse–Sard's theorem
- orbifolds, diffeological spaces — reviews welcome
- differential forms — help welcome

High-level challenges

1. boilerplate: typeclasses
   "let E be a smooth vector bundle over a smooth manifold M"

```
variable (F : Type*) [NormedAddCommGroup F] [NormedSpace 𝕜 F]
  (E : M → Type*) [TopologicalSpace (TotalSpace F E)]
  [∀ x, AddCommGroup (E x)] [∀ x, Module 𝕜 (E x)] [∀ x : M, TopologicalSpace (E x)]
  [FiberBundle F E] [VectorBundle 𝕜 F E] [ContMDiffVectorBundle n F E I]
```

2. verbosity: "let $s\colon M \to E$ be a C^n section at x"

```
variable {s : (x : M) → V x} {x : M}
  {hs : ContMDiffAt I (I.prod 𝓘(𝕜, F)) n (fun x ↦ TotalSpace.mk' F x (s x)) x}
```

3. invisible mathematics (cf. Andrej Bauer)
   use subtypes, or junk value pattern:
   lots of trivial proofs "this point lies in this open set"

High-level challenges

4. Can we find better abstractions?
   `ContinuousWithinAt`, `ContMDiffWithinAt` are a good abstraction, but very low-level
5. Better abstractions: can we abstract "this is just a local argument"? Make a tactic for this?

High-level challenges

4. Can we find better abstractions?
   `ContinuousWithinAt`, `ContMDiffWithinAt` are a good abstraction, but very low-level
5. Better abstractions: can we abstract "this is just a local argument"? Make a tactic for this?
6. API consistency vs. combinatorial explosion
   - often 4 variants of each lemma `ContMDiff{,On,At,WithinAt}`
   - mirror for `MDifferentiable*`, `ContDiff*`, `Differentiable*` (and sometimes `Continuous*`) ⇒ **20 versions**
   - applied vs non-applied: `contMDiffAt_smul` vs `contMDiffAt_fun_smul` ⇒ up to **40 versions**
   - real-life example: one PR (50 lines) became 5 (250 lines)
   - real-life example: 1000 lines of mathlib code, just copy-pasting `ContMDiff` lemmas to `MDifferentiable`

Some solutions

typeclasses: proposed solution (improved `class abbrev`) is blocked on changes to Lean itself

Some solutions

typeclasses: proposed solution (improved `class abbrev`) is blocked on changes to Lean itself

write custom elaborators to reduce the verbosity

- auto-convert section into dependent function
  `T% s` means `fun x => TotalSpace.mk' F x (s x)`

Some solutions

typeclasses: proposed solution (improved `class abbrev`) is blocked on changes to Lean itself

write custom elaborators to reduce the verbosity

- auto-convert section into dependent function
  `T% s` means `fun x => TotalSpace.mk' F x (s x)`
- infer model with corners:
  `CMDiffAt n f x` means `ContMDiffAt I J n f x`
- vision: almost never need to write the model with corners by hand

Some solutions

typeclasses: proposed solution (improved `class abbrev`) is blocked on changes to Lean itself

write custom elaborators to reduce the verbosity

- auto-convert section into dependent function
  `T% s` means `fun x => TotalSpace.mk' F x (s x)`
- infer model with corners:
  `CMDiffAt n f x` means `ContMDiffAt I J n f x`
- vision: almost never need to write the model with corners by hand
- optional: if needed, can write code as before

e.g., "X is a C^n vector field at x" becomes `(hX: CMDiffAt (T% X) x)`

Some solutions

typeclasses: proposed solution (improved `class abbrev`) is blocked on changes to Lean itself

write custom elaborators to reduce the verbosity

- auto-convert section into dependent function
  `T% s` means `fun x => TotalSpace.mk' F x (s x)`
- infer model with corners:
  `CMDiffAt n f x` means `ContMDiffAt I J n f x`
- vision: almost never need to write the model with corners by hand
- optional: if needed, can write code as before

e.g., "X is a C^n vector field at x" becomes `(hX: CMDiffAt (T% X) x)`

`fun_prop` can be supported using similar code

Some solutions (cont.)

Use tactics to auto-generate and keep APIs in sync

idea: `to_mdifferentiable` attribute

- replace `ContMDiff*` (hypotheses and goals) by the analogous `MDifferentiable` statement
- implementation very similar to `to_additive`

Some solutions (cont.)

Use tactics to auto-generate and keep APIs in sync

idea: `to_mdifferentiable` attribute

- replace `ContMDiff*` (hypotheses and goals) by the analogous `MDifferentiable` statement
- implementation very similar to `to_additive`

`to_fun` attribute
given a lemma, automatically generate the applied form
(e.g. `contMDiff_smul` to `contMDiff_fun_smul`)

Summary

1. Finding the right definitions is hard (mathematical) work.
2. Invisible mathematics matters, and becomes visible when formalising.
3. Tooling matters for making formalisation ergonomic.

Thanks for listening! Any questions?

Bonus slides

- The right definition of models with corners
- More about infinite-dimensional manifolds

The right definition of models with corners

- avoid using just Euclidean quadrants (abstraction!)
- partial equivalence from the full space
- initial requirement: range is a set of unique differentiability (to do calculus, cf. last week)

The right definition of models with corners

- avoid using just Euclidean quadrants (abstraction!)
- partial equivalence from the full space
- initial requirement: range is a set of unique differentiability (to do calculus, cf. last week)
  issue: complex manifolds are not real manifolds
- added: interior of range is dense (for symmetry of the second derivative)
- final addition: require a convex range (for $\mathbb{R}$ or $\mathbb{C}$)
  for Riemannian manifolds (length using curves matches existing metric)
- open question: complex manifolds with *complex* boundary
  (real boundary is already fine)

Infinite-dimensional manifolds

Many different definitions, trade-off between generality and ease of use.

- Hilbert manifolds
- Inverse Limit Hilbert manifolds
  e.g. $C^\infty(M, N)$ is the inverse limit of the sequence $W^{k,2}(M, N)$
- Banach manifolds
- Fréchet manifolds
- convenient calculus, Bastiani calculus, ...
  ("Diff(M) is an infinite-dimensional Lie group")

Infinite-dimensional manifolds: which one to use?

- Hilbert manifolds: smooth partitions of unity, calculus; very convenient to work with (but restricted setting)
- Banach manifolds: have calculus (but no smooth partitions of unity)

Infinite-dimensional manifolds: which one to use?

- Hilbert manifolds: smooth partitions of unity, calculus; very convenient to work with (but restricted setting)
- Banach manifolds: have calculus (but no smooth partitions of unity)

- Fréchet manifolds: lose the implicit and inverse function theorem
- convenient calculus, Bastiani calculus: non-intuitive; not mainstream

Mathlib: Banach manifolds (with completeness optional)