

# Differential geometry in Mathlib: present and future

Michael B. Rothgang (he/him)

Formalised mathematics group  
Universität Bonn



Logic and formalization seminar, Utrecht University  
June 10, 2026

# I care about many (formalization) things

- PhD in differential geometry  
(holomorphic curves in symplectic manifolds)
- formalising analysis: Carleson project (e.g. norms),  
Sobolev spaces (joint with Filippo Nuccio and Floris van Doorn)
- formalising the foundations of differential geometry
- community management, mathlib maintainer
- user tooling: linters etc.;  
mathlib's review dashboard (a.k.a. queueboard)

# I care about many (formalization) things

- PhD in differential geometry (holomorphic curves in symplectic manifolds)
- formalising analysis: Carleson project (e.g. enorms), Sobolev spaces (joint with Filippo Nuccio and Floris van Doorn)
- formalising the foundations of differential geometry
- community management, mathlib maintainer
- user tooling: linters etc.; mathlib's review dashboard (a.k.a. queueboard)

## Mathlib review and triage dashboard

Welcome to the mathlib review and triage webpage! There are many ways to help, what are you looking for in particular?

[Review queue](#)[For maintainers \(quick\)](#)[Help out](#)[Triage dashboard](#)[Why is my PR not on the queue? Can I see all my PRs?](#)

# I care about many (formalization) things

- PhD in differential geometry (holomorphic curves in symplectic manifolds)
- formalising analysis: Carleson project (e.g. enorms), Sobolev spaces (joint with Filippo Nuccio and Floris van Doorn)
- **formalising the foundations of differential geometry**
- community management, mathlib maintainer
- user tooling: linters etc.; mathlib's review dashboard (a.k.a. queueboard)

## Mathlib review and triage dashboard

Welcome to the mathlib review and triage webpage! There are many ways to help, what are you looking for in particular?

[Review queue](#)[For maintainers \(quick\)](#)[Help out](#)[Triage dashboard](#)[Why is my PR not on the queue? Can I see all my PRs?](#)

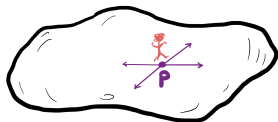
# Outline of today's talk

- 1 Basics of differential geometry
- 2 Differential geometry in mathlib
- 3 Challenges for differential geometry
- 4 Case study: formalising connections and Riemannian geometry

# Differential geometry studies smooth manifolds

smooth **manifold**: “topological space on which we can do calculus”

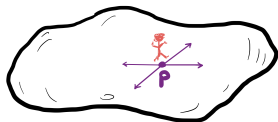
second countable Hausdorff topological space locally homeomorphic to Euclidean space, with smooth coordinate changes



# Differential geometry studies smooth manifolds

smooth **manifold**: “topological space on which we can do calculus”

second countable Hausdorff topological space locally homeomorphic to Euclidean space, with smooth coordinate changes

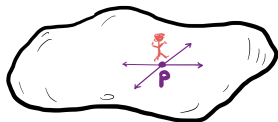


intuition: insect crawls about the earth

# Differential geometry studies smooth manifolds

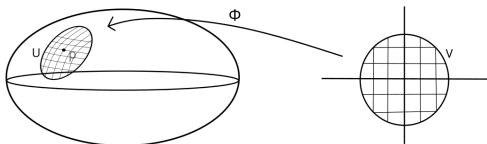
smooth **manifold**: “topological space on which we can do calculus”

second countable Hausdorff topological space locally homeomorphic to Euclidean space, with smooth coordinate changes



intuition: insect crawls about the earth

smoothness: coordinate changes between charts are smooth



# Mathlib's definition of smooth manifolds

Mathlib's definition captures all of the following features

- manifolds with boundary (e.g. interval  $[0, 1]$ )
- manifolds with corners (of any order), e.g. cube  $[0, 1]^n$

# Mathlib's definition of smooth manifolds

Mathlib's definition captures all of the following features

- manifolds with boundary (e.g. interval  $[0, 1]$ )
- manifolds with corners (of any order), e.g. cube  $[0, 1]^n$
- infinite-dimensional manifolds (e.g.  $C^k(M, N)$ )
- over different fields: over  $\mathbb{R}$ ,  $\mathbb{C}$  or  $\mathbb{Q}_p$

# Mathlib's definition of smooth manifolds

Mathlib's definition captures all of the following features

- manifolds with boundary (e.g. interval  $[0, 1]$ )
- manifolds with corners (of any order), e.g. cube  $[0, 1]^n$
- **infinite-dimensional manifolds** (e.g.  $C^k(M, N)$ )
- over different fields: over  $\mathbb{R}$ ,  $\mathbb{C}$  or  $\mathbb{Q}_p$

# Mathlib's definition of smooth manifolds

Mathlib's definition captures all of the following features

- manifolds with boundary (e.g. interval  $[0, 1]$ )
- manifolds with corners (of any order), e.g. cube  $[0, 1]^n$
- infinite-dimensional manifolds (e.g.  $C^k(M, N)$ )
- over different fields: over  $\mathbb{R}$ ,  $\mathbb{C}$  or  $\mathbb{Q}_p$

Mathlib's definition is more general than any textbook I know of.

# 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  $H \rightarrow E$   
(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  $H \rightarrow E$  (e.g. **canonical inclusion**)
- charts on  $M$  (one preferred chart at each point)
- compatibility condition: transition maps lie in structure groupoid

# Smoothness of manifolds

Hierarchy of regularity of manifolds: continuous ( $C^0$ ), continuously differentiable ( $C^n$  for  $n \in \mathbb{N}$ ), smooth ( $C^\infty$ ), analytic ( $C^\omega$ )  
All integrated in one definition.

# Smoothness of manifolds

Hierarchy of regularity of manifolds: continuous ( $C^0$ ), continuously differentiable ( $C^n$  for  $n \in \mathbb{N}$ ), smooth ( $C^\infty$ ), analytic ( $C^\omega$ )  
All integrated in one definition.

Motivation: second derivative is symmetric for  $C^2$  maps (over  $\mathbb{R}$  or  $\mathbb{C}$ ) or analytic maps (over any normed field)

# 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)

# 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)  
issue: complex manifolds are not real manifolds
- added: interior of range is dense (for symmetry of the second derivative)
- most recent addition: require a convex range (for  $\mathbb{R}$  or  $\mathbb{C}$ )  
for Riemannian manifolds (length using curves matches existing metric)

# 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)  
issue: complex manifolds are not real manifolds
- added: interior of range is dense (for symmetry of the second derivative)
- most recent 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)

# A collaborative endeavour

Major contributions by



Left to right, top to bottom: Floris van Doorn, Sébastien Gouëzel, Yury Kudryashov, Heather Macbeth, Patrick Massot, Winston Yin  
and many reviews by Oliver Nash

# Overview: differential geometry in mathlib

- general theory of smooth manifolds: allows infinite-dimension, boundaries and corners, different fields (e.g.  $\mathbb{R}$ ,  $\mathbb{C}$ ,  $p$ -adics)
- smooth maps, (continuous) differentiability
- (manifold) Fréchet derivative, chain rule

# Overview: differential geometry in mathlib

- general theory of smooth manifolds: allows infinite-dimension, boundaries and corners, different fields (e.g.  $\mathbb{R}$ ,  $\mathbb{C}$ ,  $p$ -adics)
- smooth maps, (continuous) differentiability
- (manifold) Fréchet derivative, chain rule
- products and disjoint unions of manifolds
- classification of 0-dimensional manifolds
- diffeomorphisms, local diffeomorphisms, immersions, smooth embeddings, submersions

# Overview of differential geometry (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
- affine connections

# Overview of differential geometry (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
- affine connections
  
- Lie bracket of vector fields; Lie groups and their Lie algebra
- smooth bundle metrics; Riemannian manifolds (very basic)
- examples:  $\mathbb{R}^n$ , half-space, quadrants; intervals  
unit sphere; units in Lie groups
- existence of integral curves and local flows

# In progress work (cont.)

- (R.) inverse function theorem; better characterisation of immersions and local diffeomorphisms
- (R.) immersed and embedded submanifolds
- (Naranjo) submersions, regular value theorem; master's thesis

# In progress work (cont.)

- (R.) inverse function theorem; better characterisation of immersions and local diffeomorphisms
- (R.) immersed and embedded submanifolds
- (Naranjo) submersions, regular value theorem; master's thesis
  
- oriented and orientable manifolds
- (Montero et al) quotient manifolds: partially merged; rest under review
- (Siebek) classification of closed 1-manifolds

# In progress work (cont.)

- (Scholz) collar neighborhood theorem; gluing of manifolds: master's thesis
- (Eltschig) principal fiber bundles and their classification theorem: master's thesis
- (Eltschig) orbifolds, diffeological spaces: partially merged/ under review
- (Macbeth-Lindauer-Kudryashov) differential forms: local theory merged

## In progress work (cont.)

- (Scholz) collar neighborhood theorem; gluing of manifolds: master's thesis
- (Eltschig) principal fiber bundles and their classification theorem: master's thesis
- (Eltschig) orbifolds, diffeological spaces: partially merged/ under review
- (Macbeth-Lindauer-Kudryashov) differential forms: local theory merged
- (Steinitz) existence of Riemannian metrics: under review
- (Yin) existence of local flows: partially under review
- (Hales-Lezeau-Macbeth-Massot-R.) jet spaces and jet bundles: first design discussions; no consensus yet
- (R.) unoriented bordism groups: depends on embeddings and gluing

# In progress work: Riemannian geometry

- (Macbeth-Massot-R.) Levi-Civita connection: under review
- (Macbeth-Massot-R.) Ehresmann connections: in progress
- (Macbeth-Massot-R.) curvature and geodesics: in progress

# In progress work: Riemannian geometry

- (Macbeth-Massot-R.) Levi-Civita connection: under review
  - (Macbeth-Massot-R.) Ehresmann connections: in progress
  - (Macbeth-Massot-R.) curvature and geodesics: in progress
- 
- (Tyburn) hyperbolic space and its Riemannian geometry: master's thesis

# What's missing/help wanted

- local flows: smooth dependence on initial conditions
- differential forms: deduce global definitions and theory
- Moreira's version of Sard's theorem on manifolds: polish and upstreaming
- smooth subbundles; smooth fibre bundles

# High-level challenges for formalisation

## 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 $s : M \rightarrow V$ be a $C^n$ section at $x$ ”

```

variable {s : (x : M) → V x} {x : M}
| {hs : ContMDiffAt I (I.prod J(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 for formalisation (cont.)

- 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 for formalisation (cont.)

- 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 basic computations are too painful: no support for fun\_prop
- 7 API consistency vs. combinatorial explosion
  - often 4 variants of each lemma `ContMDiff{,On,At,WithinAt}`
  - mirror for `MDifferentiable*`, `ContDiff*`, `Differentiable*` (and sometimes `Continuous*`)  $\Rightarrow$  **20 versions**
  - applied vs non-applied: `contMDiffAt_fun_smul` vs `contMDiffAt_smul`  $\Rightarrow$  **40 versions**
  - real-life example: one PR (50 lines) became 5 (250 lines)
  - real-life example: 800 lines of copy-pasting code for `ContMDiff` to `MDifferentiable`

# Solutions: typeclass boilerplate

revised `class abbrev` could address this  
needs further core Lean changes to be usable

# Solutions: domain-specific automation

write custom elaborators to reduce the verbosity

sections of vector bundles are dependent functions  $s : (x : V) \rightarrow V \times x$   
test differentiability by post-composing with inclusion into total space  
`Bundle.TotalSpace F V`

the `T%` elaborator automates this:

`T% s` means `fun x => TotalSpace.mk' F x (s x)`

## Solutions: domain-specific automation (cont.)

note: cannot make implicit, a typeclass argument or an outparam infer the model with corners automatically, when unambiguous via a custom elaborator

`CmdiffAt n f x` means `ContMDiffAt I J n f x`,  
`MdiffAt [s] f x` means `MDifferentiableWithinAt I J f s x`,

e.g., “ $X$  is a  $C^n$  vector field at  $x$ ” becomes `(hX: CmdiffAt (T% X) x)`

# Automation using custom elaborators

- support all model constructions
- custom elaborator for each definition
- matching delaborators: infoview shows `MDiff f` with hovers, not `MDifferentiable I J f`
- make extensible, supporting downstream projects
- fast(er) than writing by hand

# Automation using custom elaborators

- support all model constructions
- custom elaborator for each definition
- matching delaborators: infoview shows `MDiff f` with hovers, not `MDifferentiable I J f`
- make extensible, supporting downstream projects
- fast(er) than writing by hand

Vision: almost never have to write the model with corners by hand.  
If ambiguous, Lean asks “did you mean X or Y” with concrete suggestion

# Automation using custom elaborators

- support all model constructions
- custom elaborator for each definition
- matching delaborators: infoview shows `MDiff f` with hovers, not `MDifferentiable I J f`
- make extensible, supporting downstream projects
- fast(er) than writing by hand

Vision: almost never have to write the model with corners by hand.  
If ambiguous, Lean asks “did you mean X or Y” with concrete suggestion

`fun_prop` can be supported using similar code

# What can we do: combinatorial explosion

Use tactics to auto-generate and keep APIs in sync

`to_fun` attribute

given a lemma, automatically generate the eta-expanded form (e.g. `contMDiff_smul` to `contMDiff_fun_smul`)

idea: `to_mdifferentiable` attribute

- replace `ContMDiff` (hypotheses and goals) by the analogous `MDifferentiable` statement
- new name: straightforward replacement
- proofs: just obvious translation; if it fails, indicates missing API
- implementation is a `to_additive` extension (allow omitting arguments)

Help is welcome!

# Review: Riemannian manifolds

Riemannian manifold  $(M, g)$ : smooth manifold  $M$  with  
Riemannian metric  $g = (g_p)_{p \in M}$ ,  $g_p: T_p M \times T_p M \rightarrow \mathbb{R}$  inner product  
(varying smoothly with  $p$ )

a metric induces curvature and geodesics (locally shortest paths)

# Motivation: connections on vector bundles

$E \rightarrow M$  smooth vector bundle,  $s: M \rightarrow E$  smooth section,  
 $X$  vector field on  $M$

## Question

What is the directional derivative  $ds(x)X_x$ ?

The answer should respect addition and scalar multiplication on sections!

# Connections: three different definitions

$s: M \rightarrow E$  smooth section of  $E \rightarrow M$ ;  $X$  vector field on  $M$

What is the directional derivative  $ds(x)X_x$ ?

- ① parallel transport  
isomorphisms  $E_x \rightarrow E_{\gamma(t)}$  for any smooth path  $\gamma(t) \in M$
- ② Ehresman connection  
splitting  $TE = HE \oplus VE$  into horizontal and vertical sub-bundles,  
where  $VE$  are all vectors tangent to any fiber
- ③ Koszul connection  
 $\Gamma(E) \rightarrow \Gamma(\text{Hom}(TM, E))$  depends on  $X$  only via  $X_x$ , i.e. induces  
 $(\prod_x : M, E_x) \rightarrow (\prod_x : M, \text{TangentSpace } I_x \rightarrow L[\mathbb{R}] E_x)$

## Fact

Each Riemannian manifold  $(M, g)$  has a unique connection which is compatible with  $g$  and torsion-free, called the **Levi-Civita connection**.

# Formalisation in Lean: overview and some statistics

- 3000 lines of code merged to mathlib, another 4000 pending
- custom elaborators: another 1000 lines (+2000 lines of tests)
  
- Koszul connections; classification over trivial bundles (and connection 1-forms)
- existence and uniqueness of Levi-Civita connections
- induced Ehresman connection
- definition of geodesics

Next steps: smoothness of geodesics, local existence  
pullback of Ehresman connections, parallel transport

# Koszul connections in Lean

```
/-- A function from sections of a vector bundle `V` on a manifold `M` to sections of $Hom(TM, E)$
is a covariant derivative over a set `s` in `M` if it is additive and satisfies the Leibniz rule
when applied to sections that are differentiable at a point of `s`.
```

```
Caution, the argument order is nonstandard: `cov σ x (X x)` corresponds to `∇_X σ x` on paper. -/
structure IsCovariantDerivativeOn
```

```
  (cov : (Π x : M, V x) → (Π x : M, TangentSpace I x →L[k] V x))
  (s : Set M := Set.univ) : Prop where
add {σ σ' : Π x : M, V x} {x}
  (hσ : MDiffAt (T% σ) x) (hσ' : MDiffAt (T% σ') x) (hx : x ∈ s := by trivial) :
  cov (σ + σ') x = cov σ x + cov σ' x
leibniz {σ : Π x : M, V x} {g : M → k} {x}
  (hσ : MDiffAt (T% σ) x) (hg : MDiffAt g x) (hx : x ∈ s := by trivial) :
  cov (g • σ) x = g x • cov σ x + (d% g x).smulRight (σ x)
```

# Koszul connections in Lean

```
/-- A function from sections of a vector bundle `V` on a manifold `M` to sections of  $\text{Hom}(TM, E)$ 
is a covariant derivative over a set `s` in `M` if it is additive and satisfies the Leibniz rule
when applied to sections that are differentiable at a point of `s`.
```

```
Caution, the argument order is nonstandard: `cov σ x (X x)` corresponds to  $\nabla_X \sigma x$  on paper. -/
structure IsCovariantDerivativeOn
```

```
  (cov : (Π x : M, V x) → (Π x : M, TangentSpace I x →L[k] V x))
  (s : Set M := Set.univ) : Prop where
add {σ σ' : Π x : M, V x} {x}
  (hσ : MDiffAt (T% σ) x) (hσ' : MDiffAt (T% σ') x) (hx : x ∈ s := by trivial) :
  cov (σ + σ') x = cov σ x + cov σ' x
leibniz {σ : Π x : M, V x} {g : M → k} {x}
  (hσ : MDiffAt (T% σ) x) (hg : MDiffAt g x) (hx : x ∈ s := by trivial) :
  cov (g • σ) x = g x • cov σ x + (d% g x).smulRight (σ x)
```

## Notable features of this definition

- unbundled, localised to a set (also have global bundled definition)
- auto-parameters: specializes ergonomically to `univ`

# Koszul connections in Lean

```
/-- A function from sections of a vector bundle `V` on a manifold `M` to sections of  $\text{Hom}(TM, E)$ 
is a covariant derivative over a set `s` in `M` if it is additive and satisfies the Leibniz rule
when applied to sections that are differentiable at a point of `s`.
```

Caution, the argument order is nonstandard: `cov σ x (X x)` corresponds to  $\nabla_X \sigma x$  on paper. -/

```
structure IsCovariantDerivativeOn
  (cov : (Π x : M, V x) → (Π x : M, TangentSpace I x →L[k] V x))
  (s : Set M := Set.univ) : Prop where
add {σ σ' : Π x : M, V x} {x}
  (hσ : MDiffAt (T% σ) x) (hσ' : MDiffAt (T% σ') x) (hx : x ∈ s := by trivial) :
  cov (σ + σ') x = cov σ x + cov σ' x
leibniz {σ : Π x : M, V x} {g : M → k} {x}
  (hσ : MDiffAt (T% σ) x) (hg : MDiffAt g x) (hx : x ∈ s := by trivial) :
  cov (g • σ) x = g x • cov σ x + (d% g x).smulRight (σ x)
```

## Notable features of this definition

- unbundled, localised to a set (also have global bundled definition)
- auto-parameters: specializes ergonomically to `univ`
- note the custom elaborators
- scalar multiplication and `d%`

# Koszul connections in Lean

```
/-- A function from sections of a vector bundle `V` on a manifold `M` to sections of  $\text{Hom}(TM, E)$ 
is a covariant derivative over a set `s` in `M` if it is additive and satisfies the Leibniz rule
when applied to sections that are differentiable at a point of `s`.
```

Caution, the argument order is nonstandard: `cov σ x (X x)` corresponds to  $\nabla_X \sigma x$  on paper. -/

```
structure IsCovariantDerivativeOn
  (cov : (Π x : M, V x) → (Π x : M, TangentSpace I x →L[k] V x))
  (s : Set M := Set.univ) : Prop where
add {σ σ' : Π x : M, V x} {x}
  (hσ : MDiffAt (T% σ) x) (hσ' : MDiffAt (T% σ') x) (hx : x ∈ s := by trivial) :
  cov (σ + σ') x = cov σ x + cov σ' x
leibniz {σ : Π x : M, V x} {g : M → k} {x}
  (hσ : MDiffAt (T% σ) x) (hg : MDiffAt g x) (hx : x ∈ s := by trivial) :
  cov (g • σ) x = g x • cov σ x + (d% g x).smulRight (σ x)
```

## Notable features of this definition

- unbundled, localised to a set (also have global bundled definition)
- auto-parameters: specializes ergonomically to `univ`
- note the custom elaborators
- **scalar multiplication and  $d\%$**

# Stating the Leibniz rule correctly

Leibniz rule on paper ( $g: M \rightarrow \mathbb{R}$ ,  $\sigma$  section of  $E$ ):  
 $\nabla_X(g \bullet \sigma)_x = g_x \bullet \nabla_X \sigma + dg_x \bullet \sigma(x)$  for all  $x$

caution: `mfderiv% g x • σ x` does not type-check,  
 $dg_x$  maps into  $T_{g(x)}\mathbb{R}$  (not  $\mathbb{R}$ !)

solution 1: force defeq abuse through type ascription

solution 2: explicit identification  $T_x\mathbb{R} \cong \mathbb{R}$

final solution: combine `mfderiv` with that identification

→ `mvfderiv TxM → ℝ`, notation `d% g`

similarly, `mvfderivWithin` with notation `d[s] g`

# Stating the chain rule for `mderiv` correctly

$f: M \rightarrow k$  and  $g: M \rightarrow E$  differentiable

chain rule for `mderiv`:  $d(f \bullet g)(v) = f \bullet dg(v) + df(v) \bullet g$

# Stating the chain rule for `mfderiv` correctly

$f: M \rightarrow k$  and  $g: M \rightarrow E$  differentiable

chain rule for `mfderiv`:  $d(f \bullet g)(v) = f \bullet dg(v) + df(v) \bullet g$

Note: left hand side lives in  $T_{(f \bullet g)v} E$ , right hand side in  $T_{g(v)} E$

Demo: four different versions in Lean — for simp confluence

# Stating the chain rule for `mfderiv` correctly

$f: M \rightarrow k$  and  $g: M \rightarrow E$  differentiable

chain rule for `mfderiv`:  $d(f \bullet g)(v) = f \bullet dg(v) + df(v) \bullet g$

Note: left hand side lives in  $T_{(f \bullet g)v} E$ , right hand side in  $T_{g(v)} E$

Demo: four different versions in Lean — for simp confluence

Real fix: use `mvfderiv` instead, and just have one lemma!

# Making automation work

## A fragment of the Levi-Civita file

```

/-- `leviCivitaAux` is tensorial with respect to its second argument. -/
theorem leviCivitaAux_tensorial2 [FiniteDimensional ℝ E]
  {Y : Π x : M, TangentSpace I x} (x : M) (hY : MDiffAt (T% Y) x) {X : Π x, TangentSpace I x}
  (hX : MDiffAt (T% X) x) :
  TensorialAt I E (leviCivitaAux I X Y · x) x where
smul hf hZ := by
  simp (disch := fun_prop) [leviCivitaAux,
    mlieBracket_smul_right, mlieBracket_smul_left,
    mvfderiv_fun_mul,
    inner_smul_left, inner_smul_right, inner_add_right, real_inner_comm]
  ring

```

- custom elaborators are great
- simp works — defeq abuse is fixed

# Making automation work

## A fragment of the Levi-Civita file

```

/-- `leviCivitaAux` is tensorial with respect to its second argument. -/
theorem leviCivitaAux_tensorial₂ [FiniteDimensional ℝ E]
  {Y : Π x : M, TangentSpace I x} (x : M) (hY : MDiffAt (T% Y) x) {X : Π x, TangentSpace I x}
  (hX : MDiffAt (T% X) x) :
  TensorialAt I E (leviCivitaAux I X Y · x) x where
smul hf hZ := by
  simp (disch := fun_prop) [leviCivitaAux,
    mlieBracket_smul_right, mlieBracket_smul_left,
    mvfderiv_fun_mul,
    inner_smul_left, inner_smul_right, inner_add_right, real_inner_comm]
  ring

```

- custom elaborators are great
- simp works — defeq abuse is fixed
- fun\_prop discharger — prototypical support for manifolds

# Honourable mention: a bug in Lean

```
variable {I} in
@[fun_prop] lemma _root_.MDifferentiable.inner_bundle' {X Y :  $\Pi$  x : M, TangentSpace I x}
  (hX : MDiff (T% X)) (hY : MDiff (T% Y)) : MDiff  $\langle$ X, Y $\rangle$  :=
  MDifferentiable.inner_bundle hX hY

variable {I} in
@[fun_prop] lemma _root_.MDifferentiableAt.inner_bundle' {x : M} {X Y :  $\Pi$  x : M, TangentSpace I x}
  (hX : MDiffAt (T% X) x) (hY : MDiffAt (T% Y) x) :
  MDiffAt  $\langle$ X, Y $\rangle$  x :=
  MDifferentiableAt.inner_bundle hX hY
```

Special-cased copy of a lemma, to avoid instance synthesis failures

Root cause: typeclass synthesis seeing through type synonyms  
(TangentSpace I x versus E)

Similar to Lean's issue 9077, requires new Lean core fix

# Summary

- ① Finding the right definitions is hard (mathematical) work.
- ② Invisible mathematics matters, and becomes visible when formalising.
- ③ Definitional equality abuse comes back to haunt you.  
Fixing it will simplify your life in unexpected ways.
- ④ Tooling matters for making formalisation ergonomic.

# Summary

- ① Finding the right definitions is hard (mathematical) work.
- ② Invisible mathematics matters, and becomes visible when formalising.
- ③ Definitional equality abuse comes back to haunt you.  
Fixing it will simplify your life in unexpected ways.
- ④ Tooling matters for making formalisation ergonomic.

Thanks for listening! Any questions?

# Bonus slides

- ▶ More details on infinite-dimensional manifolds
- ▶ Why such a general definition?
- ▶ Why do we need elaborators for the models with corners an outparam

# 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, ...  
 (“ $\text{Diff}(M)$  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)

# Why do we care about this general manifolds?

- one definition to rule them all
- manifolds with boundary have many applications:  
differential topology, bordism theory, symplectic filling problems,  
connected sums, handle attachments, surgery

# Why do we care about this general manifolds?

- one definition to rule them all
- manifolds with boundary have many applications: differential topology, bordism theory, symplectic filling problems, connected sums, handle attachments, surgery
- manifolds with corners: closure under products (and because you can!)
- infinite-dimensional manifolds: study manifolds of maps  
e.g. smoothness of geodesics (cf. Lang), moduli space of holomorphic curves in symplectic geometry

# Products and the model with corners

- manifolds with corners are closed under products
- detail: product of normed spaces has two models with corners

# Products and the model with corners

- manifolds with corners are closed under products
- detail: product of normed spaces has two models with corners  
 models are equal (“can prove this”), but not by definition
- need to specify the model with corners,  
 cannot let e.g. typeclass inference guess it