Geometry in Lean: a Report for Mathematicians

Nicolò Cavalleri
Joint work with Anthony Bordg

31 July 2021

CICM 2021 - FMM

Overview

Geometry with proof assistants might feel harder than other areas of mathematics because:

- In proofs many details are skipped
- Sometimes proofs in books rely too much on intuition and are not formal enough
- Geometrical objects often carry with them a lot of data

Lean's progress in geometry

Differential geometry is just at the beginning, but Lean is doing considerable progress.

- **Isabelle** has smooth manifolds, smooth maps, tangent and cotangent spaces, spheres and projective spaces, but does not have the tangent bundle.
- **Coq** does not appear to have any formalization of standard differential geometry, although it has a stub of synthetic differential geometry.

Lean's progress in geometry

Lean at the moment has:

- Smooth manifolds with corners over arbitrary nondiscrete normed fields
- Tangent spaces and tangent bundles
- Smooth maps and diffeomorphisms
- Partitions of unity
- Lie groups and an implementation of their Lie algebras (at least in the real case)
- The sphere
- A primitive version of the Whitney embedding theorem

Lean's future in geometry

In the next month or so Lean should have:

- Smooth fiber and vector bundles and their operations
- Cotangent bundle
- Vector fields and differential forms
- A newer definition of the Lie algebra of a Lie group
- Hopefully the definition of Riemannian manifolds

Smooth manifolds with corners

Smooth manifolds with corners

Manifolds in Lean have been formalized by Sébastien Gouëzel.

They are defined very generally: we have smooth manifolds with corners over an arbitrary nondiscrete normed field.

Given a field $\mathbb{k}$, a **norm** is a real valued map $|\cdot| : \mathbb{k} \to \mathbb{R}$, that respects the following conditions:

$$\forall x\ y \in \mathbb{k}, |xy| = |x||y|$$

$$\forall x\ y \in \mathbb{k}, |x + y| \leq |x| + |y|$$

In the literature usually, such a map is called a valuation, but in Lean this naming convention was used.

A normed field is **nondiscrete** if there exists a $x \in \mathbb{k}$ such that

$$|x| < 1$$

Smooth manifolds with corners

A model with corners is a map $I : H \to E$ where

- H is a topological space
- E is a normed vector space over $\mathbb{k}$
- I is an embedding
- the image of I is nice, i.e. at each point of its image the tangent cone spans a dense subset of the whole space

```
structure model_with_corners (𝕜 : Type*) [nondiscrete_normed_field 𝕜]
  (E : Type*) [normed_group E] [normed_space 𝕜 E]
  (H : Type*) [topological_space H]
extends local_equiv H E :=
(source_eq          : source = univ)
(unique_diff'       : unique_diff_on 𝕜 to_local_equiv.target)
(continuous_to_fun  : continuous to_fun . tactic.interactive.continuity')
(continuous_inv_fun : continuous inv_fun . tactic.interactive.continuity')
```

Smooth manifolds with corners

A charted space is a topological space M endowed with an atlas, i.e. a set of local homeomorphisms taking values in a model space H, called charts, such that the domains of the charts cover the whole space M.

```
class charted_space (H : Type*) [topological_space H]
  (M : Type*) [topological_space M] :=
(atlas []                : set (local_homeomorph M H))
(chart_at []            : M → local_homeomorph M H)
(mem_chart_source [] : ∀x, x ∈ (chart_at x).source)
(chart_mem_atlas []   : ∀x, chart_at x ∈ atlas)
```

Smooth manifolds with corners

A smooth manifold with corners is a charted space over H with a model with corners $I : H \to E$ for some vector space E such that the change of coordinates are smooth when read through I.

```
class smooth_manifold_with_corners {𝕜 : Type*} [nondiscrete_normed_field 𝕜]
  {E : Type*} [normed_group E] [normed_space 𝕜 E]
  {H : Type*} [topological_space H] (I : model_with_corners 𝕜 E H)
  (M : Type*) [topological_space M] [charted_space H M] extends
has_groupoid M (times_cont_diff_groupoid ∞ I) : Prop
```

Lie groups

We can go on and get smooth monoids and Lie groups:

```
class lie_group {𝕜 : Type*} [nondiscrete_normed_field 𝕜]
  {H : Type*} [topological_space H]
  {E : Type*} [normed_group E] [normed_space 𝕜 E]
  (I : model_with_corners 𝕜 E H)
  (G : Type*) [group G] [topological_space G] [charted_space H G]
extends has_smooth_mul I G : Prop :=
(smooth_inv : smooth I I (λ a:G, a⁻¹))
```

Lie groups

One problem we met in the formalization of Lie groups comes from the fact that the maps

$$(\mathrm{id} : V \to V) \times (\mathrm{id} : W \to W)$$

and

$$\mathrm{id} : V \times W \to V \times W$$

are not **definitionally equal**. This puts a constraint on the definition of Lie groups in order to make them stable under product: we have to allow them to have borders.

Lie groups

Two types that are equal by definition are said to be **definitionally equal**. If you need a proper proof to see that they are equal, then they are not definitionally equal, and Lean will not know they are equal unless you tell it where to find the proof.

Vector bundles

Vector bundle

In standard mathematics, a vector bundle is given by

- two topological spaces Z and B
- a continuous surjection $\pi : Z \to B$
- a structure of vector space on each fiber of the projection π
- a *model* vector space V

that satisfy

- for any p in B, there exists a neighborhood U of p such that $U \times V$ and $\pi^{-1}(U)$ are homeomorphic through a map ϕ
- ϕ is such that $\pi \circ \phi$ is the projection onto the first factor
- ϕ is a linear isomorphism fiberwise

Vector bundle

The first thing that comes to mind when reproducing this in Lean is to have indeed two topological spaces Z and B:

```
variables {B : Type*} [topological_space B]
  {Z : Type*} [topological_space Z]
```

and a projection `proj : Z → B`. How do we now put a vector space structure on the fibers (`{y // proj y = x}`)? We just quantify instances:

```
variables [normed_field 𝕜] [∀ (x : B), add_comm_group {y // proj y = x}]
  [∀ (x : B), module 𝕜 {y // proj y = x}]
```

and hence, given a model fiber F, we can define trivializations

```
structure vector_bundle_trivialization
extends fiber_bundle_trivialization F proj :=
(linear : ∀ x ∈ base_set, is_linear_map 𝕜 (λ y : proj ⁻¹' {x}, (to_fun y).2))
```

Vector bundle

There is a big problem with this naif implementation: in the case of the tangent bundle, the total space was defined as the product $B \times E$ of the base space and of the model vector space for the manifold.

In a case like this, the tangent space at a point and the fiber of the projection at the same point

$$E \quad \longleftrightarrow \quad \texttt{proj}^{-1\prime}\,\{\texttt{x}\}$$

are not definitionally equal.

Vector bundle

The solution to this problem that we adopted was to force vector bundles to be implemented with Σ **types**. Σ types are dependent types that are analogous to products but where the second factor can take values in different types depending on the value of the first factor.

Virtually all vector bundles can be implemented through a Σ type.

Vector bundle

A product can be seen as a constant sigma type and indeed the type underlying the tangent bundle was changed from $B \times E$ to $\Sigma x : B, E$.

The key idea is to have this type `E : B → Type*` that given a point `x : B` gives us the fiber at that point `E x`.

So we start with the following data:

```
variables (R : Type*) {B : Type*} (F : Type*) (E : B → Type*)
  [semiring R] [∀ x, add_comm_monoid (E x)] [∀ x, module R (E x)]
  [topological_space F] [add_comm_monoid F] [module R F]
  [topological_space (total_space E)] [topological_space B]
```

Vector bundle

The projection is defined automatically as `sigma.fst`, which is another way to write λ `x, x.1`.

Note that it is still the case that `proj` $^{-1}$' $\{$`x`$\}$ is not definitionally equal to the fiber `E x`, but now that we have this strong constraint on the form of a vector bundle we do not care anymore, because we have a convenient way to talk about fibers that works well.

The end

Announcements

The paper **Elements of Differential Geometry in Lean, a Report for Mathematicians** will soon be available on arXiv and in the conference proceedings.

The paper contains more material and details, taking equality as a leitmotiv to describe the difficulties met during the formalization and how we overcame them.