Journées 2023 du GT SCALP ($\in$ GDR IM)
Orléans

Formalization of Applied Mathematics: a journey

Sylvie Boldo

Inria, Université Paris-Saclay

November 27th, 2023

My journey

In my journey, I was lucky to travel with

- François Clément,
- Florian Faissole,
- Vincent Martin,
- Micaela Mayero,
- Houda Mouhcine.

My journey

In my journey, I was lucky to travel with

- François Clément,
- Florian Faissole,
- Vincent Martin,
- Micaela Mayero,
- Houda Mouhcine.

In my other life,
I have been busy with computer arithmetic and *agrégation d'informatique*.
Do not hesitate to reach me for one of these topics.

Outline

Introduction

Mathematics

$\mathbb{R}$, $\int$, $\frac{\partial^2 u}{\partial t^2}$
theorems

Introduction

$\mathfrak{Mathematics}$	$\mathbb{R}$, $\int$, $\frac{\partial^2 u}{\partial t^2}$ theorems
Applied Mathematics	numerical scheme, convergence algorithms + theorems

Introduction

$\mathfrak{Mathematics}$	$\mathbb{R}$, $\int$, $\frac{\partial^2 u}{\partial t^2}$ theorems
Applied Mathematics	numerical scheme, convergence algorithms + theorems
Computer	floating-point numbers, implementation programs + ?

Mathe

$\mathbb{R}$, $\int$, $\frac{\partial^2 u}{\partial t^2}$

heorems

formal proof

Applie

convergence

Computer

ntation

progr

Motivations

PDE (Partial Differential Equation) $\Rightarrow$ weather forecast
$\Rightarrow$ nuclear simulation
$\Rightarrow$ optimal control
$\Rightarrow$ ...

Motivations

PDE (Partial Differential Equation) $\Rightarrow$ weather forecast
$\Rightarrow$ nuclear simulation
$\Rightarrow$ optimal control
$\Rightarrow$ ...

Usually too complex to solve by an exact mathematical formula
$\Rightarrow$ approximated by numerical scheme over discrete grids/volumes

$\Rightarrow$ mathematical proofs of the convergence of the numerical scheme
(we compute something close to the PDE solution if the size decreases)

Motivations

PDE (Partial Differential Equation) $\Rightarrow$ weather forecast
$\Rightarrow$ nuclear simulation
$\Rightarrow$ optimal control
$\Rightarrow$ ...

Usually too complex to solve by an exact mathematical formula
$\Rightarrow$ approximated by numerical scheme over discrete grids/volumes

$\Rightarrow$ mathematical proofs of the convergence of the numerical scheme
(we compute something close to the PDE solution if the size decreases)

$\Rightarrow$ real program implementing the scheme/method

Motivations

PDE (Partial Differential Equation) ⇒ weather forecast
⇒ nuclear simulation
⇒ optimal control
⇒ ...

Usually too complex to solve by an exact mathematical formula
⇒ approximated by numerical scheme over discrete grids/volumes

⇒ mathematical proofs of the convergence of the numerical scheme
(we compute something close to the PDE solution if the size decreases)

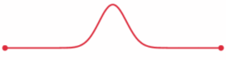
⇒ real program implementing the scheme/method

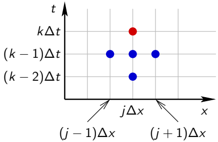
Let us machine-check this kind of programs!

1D wave equation resolution by the 3-point scheme

$$\frac{\partial^2 u(x,t)}{\partial t^2} - c^2 \frac{\partial^2 u(x,t)}{\partial x^2} = s(x,t)$$

$$\frac{u_j^k - 2u_j^{k-1} + u_j^{k-2}}{\Delta t^2} - c^2 \frac{u_{j+1}^{k-1} - 2u_j^{k-1} + u_{j-1}^{k-1}}{\Delta x^2} = s_j^{k-1}$$






```
du = u[i+1][k] - 2 * u[i][ k]  + u[i-1][k];
u[i][ k+1] = 2 * u[i][k]  - u[i][k-1] + a * du;
```

Rounding error: $\left|u_i^k - exact\left(u_i^k\right)\right| \leqslant 78 \times 2^{-53} \times (k+1) \times (k+2)$

Method error: $\left\|e_h^{k_{\Delta t}(t)}\right\|_{\Delta x} = O_{\substack{t \in [0, t_{\max}], (\Delta x, \Delta t) \to 0 \\ 0 < \Delta x \,\wedge\, 0 < \Delta t \,\wedge\, c\frac{\Delta t}{\Delta x} \leqslant 1 - \xi}} (\Delta x^2 + \Delta t^2)$

Program error: no illicit memory access, no division by zero, no overflow...

Formal verification of 32 lines of C code + 154 lines of annotations
$\hookrightarrow$ 150 theorems to prove (incl. 33 Coq theorems Coq for 15 000 lines)

Done with F. Clément, J.-C. Filliâtre, M. Mayero, G. Melquiond, P. Weis

$\Rightarrow$ now what?

Outline

Motivations

http://www.ima.umn.edu/~arnold/disasters/sleipner.html

The sinking of the Sleipner A offshore platform

Excerpted from a report of SINTEF, Civil and Environmental Engineering:

The Sleipner A platform produces oil and gas in the North Sea and is supported on the seabed at a water depth of 82 m. It is a Condeep type platform with a concrete gravity base structure consisting of 24 cells and with a total base area of 16 000 m^2. Four cells are elongated to shafts supporting the platform deck. The first concrete base structure for Sleipner A sprang a leak and sank under a controlled ballasting operation during preparation for deck mating in Gandsfjorden outside Stavanger, Norway on 23 August 1991.

Immediately after the accident, the owner of the platform, Statoil, a Norwegian oil company appointed an investigation group, and SINTEF was contracted to be the technical advisor for this group.

The investigation into the accident is described in 16 reports...

The conclusion of the investigation was that the loss was caused by a failure in a cell wall, resulting in a serious crack and a leakage that the pumps were not able to cope with. The wall failed as a result of a combination of a serious error in the finite element analysis and insufficient anchorage of the reinforcement in a critical zone.

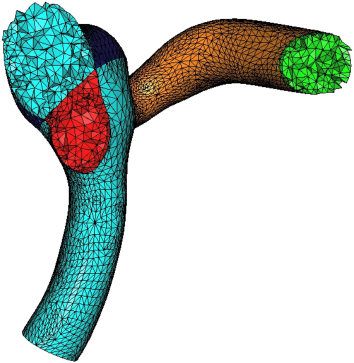
A better idea of what was involved can be obtained from this photo and sketch of the platform. The top deck weighs 57,000 tons, and provides accommodation for about 200 people and support for drilling equipment weighing about 40,000 tons. When the first model sank in August 1991, the crash

Motivations

Real life applications need solving PDE (Partial Differential Equation) on complex 3D geometries.

Motivations

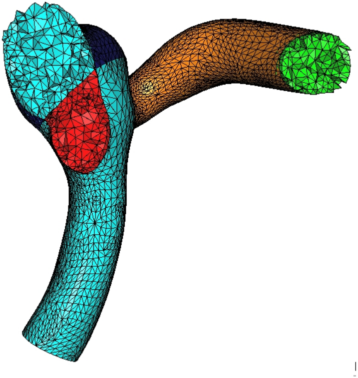
Real life applications need solving PDE (Partial Differential Equation) on complex 3D geometries.



© V. Martin

Motivations

Real life applications need solving PDE (Partial Differential Equation) on complex 3D geometries.



© V. Martin

Instead of regular 2D/3D grids, we consider meshes made of triangles/tetrahedra.

Motivations

The Finite Element Method (FEM) is the most used method to solve PDEs over meshes.

> *FEM encompasses methods for connecting many simple element equations over many small subdomains, named finite elements, to approximate a more complex equation over a larger domain.*
>
> (`https://en.wikipedia.org/wiki/Finite_element_method`)

Motivations

The Finite Element Method (FEM) is the most used method to solve PDEs over meshes.

> *FEM encompasses methods for connecting many simple element equations over many small subdomains, named finite elements, to approximate a more complex equation over a larger domain.*
>
> (https://en.wikipedia.org/wiki/Finite_element_method)

⇒ mathematical proofs of the FEM
⇒ C++ library (Felisce) implementing the FEM

Motivations

The Finite Element Method (FEM) is the most used method to solve PDEs over meshes.

> *FEM encompasses methods for connecting many simple element equations over many small subdomains, named finite elements, to approximate a more complex equation over a larger domain.*
>
> (https://en.wikipedia.org/wiki/Finite_element_method)

$\Rightarrow$ mathematical proofs of the FEM
$\Rightarrow$ C++ library (Felisce) implementing the FEM

Let us machine-check this program!

Motivations

The Finite Element Method (FEM) is the most used method to solve PDEs over meshes.

> *FEM encompasses methods for connecting many simple element equations over many small subdomains, named finite elements, to approximate a more complex equation over a larger domain.*
>
> (https://en.wikipedia.org/wiki/Finite_element_method)

$\Rightarrow$ mathematical proofs of the FEM
$\Rightarrow$ C++ library (Felisce) implementing the FEM

Let us machine-check this program!

First, let us understand/formally prove the mathematics.

Outline

Lax-Milgram?

- considered as the foremost theorem for the correctness of the Finite Element Method. (stating in a few slides)
- means that the (method) error may bounded when approximating an infinite-dimensional space by a finite-dimensional one.

  Example: functions and polynomials.

How to attack non-trivial mathematics?

Mathematicians at work for Lax-Milgram theorem

- more 50 pages of mathematical proofs

Mathematicians at work for Lax-Milgram theorem

- more 50 pages of mathematical proofs
- very detailed!

Mathematicians at work for Lax-Milgram theorem

- more 50 pages of mathematical proofs
- very detailed!
- more than 7,000 lines and 220,000 characters

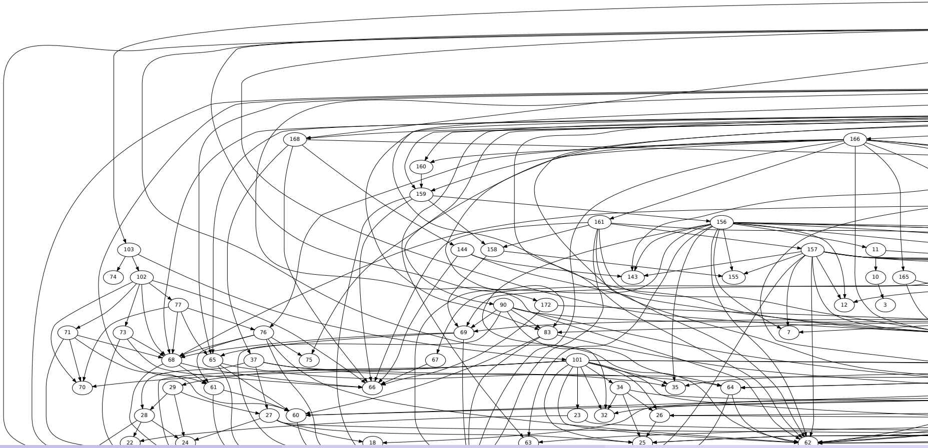
Mathematicians at work for Lax-Milgram theorem

- more 50 pages of mathematical proofs
- very detailed!
- more than 7,000 lines and 220,000 characters
- with dependencies!

Mathematicians at work for Lax-Milgram theorem

- more 50 pages of mathematical proofs
- very detailed!
- more than 7,000 lines and 220,000 characters
- with dependencies!

- more 50 pages of mathematical proofs
- very detailed!
- more than 7,000 lines and 220,000 characters
- with dependencies!



Proof engineering

Let us build upon the Coquelicot library (Boldo, Lelay, Melquiond)

+ general spaces

Proof engineering

Let us build upon the Coquelicot library (Boldo, Lelay, Melquiond)

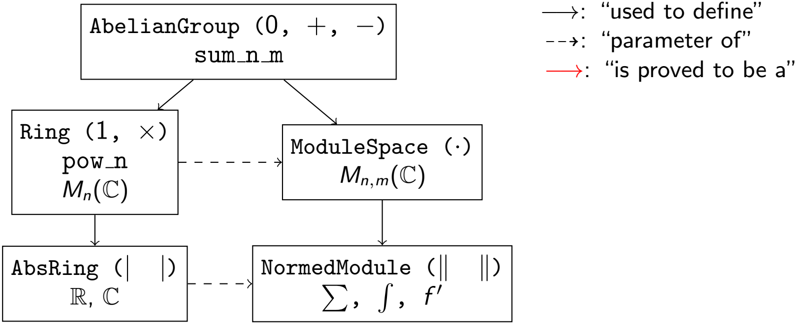
+ general spaces
+ many existing theorems

Proof engineering

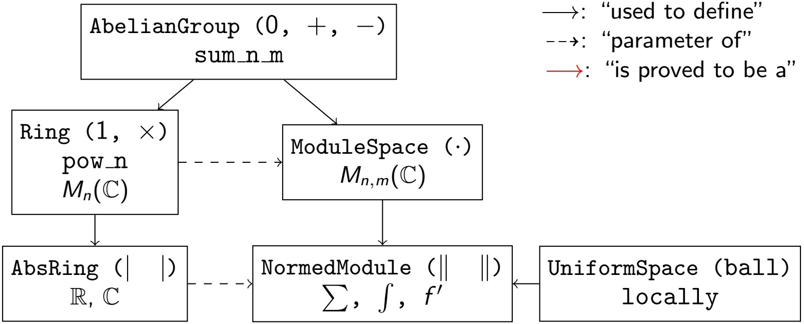
Let us build upon the Coquelicot library (Boldo, Lelay, Melquiond)

+ general spaces
+ many existing theorems
- not always the space we need

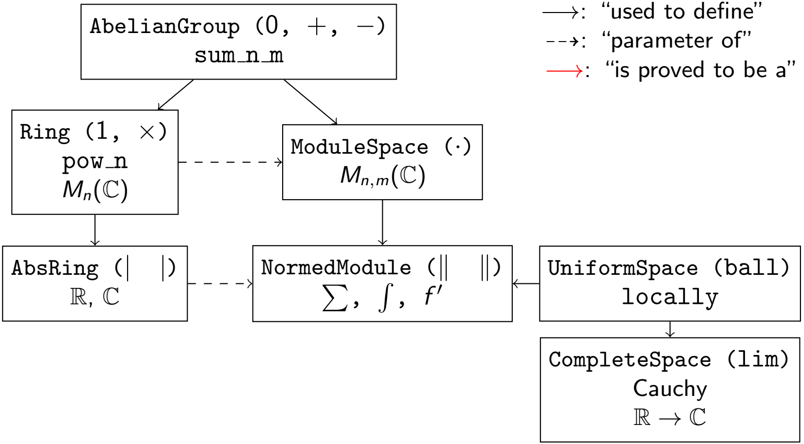
Enriched Hierarchy



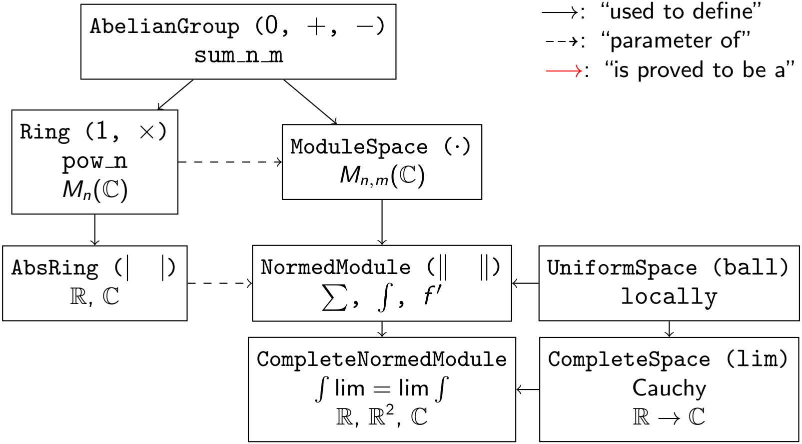
Enriched Hierarchy



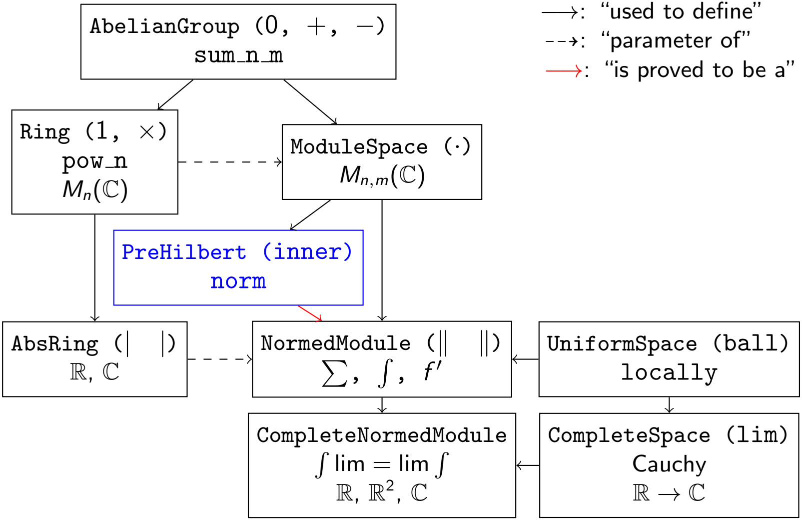
Enriched Hierarchy



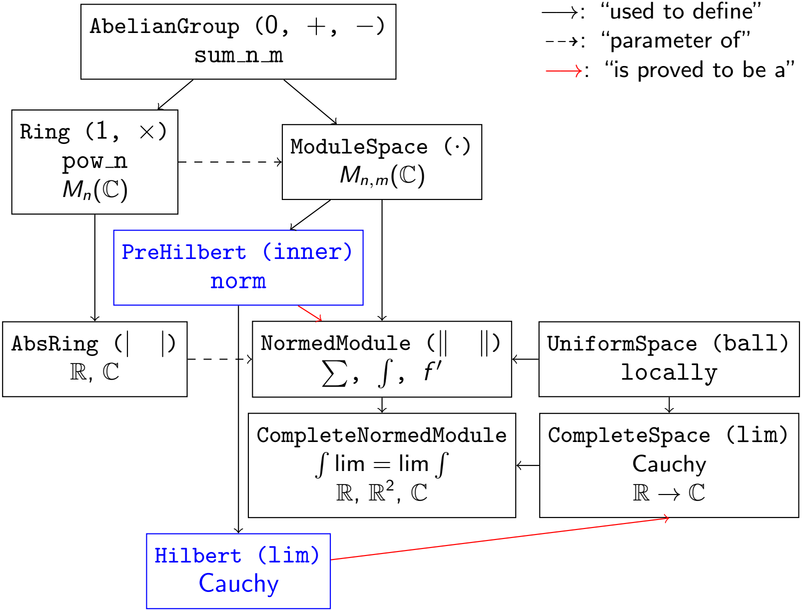
Enriched Hierarchy



Enriched Hierarchy



Enriched Hierarchy



Summary of the work done

- results about functional spaces, linear and bilinear mappings

Summary of the work done

- results about functional spaces, linear and bilinear mappings
- fixed-point theorem in a sub-complete normed module

Summary of the work done

- results about functional spaces, linear and bilinear mappings
- fixed-point theorem in a sub-complete normed module
- decide if a space ($\geq$ NormedModule) is only zero

Summary of the work done

- results about functional spaces, linear and bilinear mappings
- fixed-point theorem in a sub-complete normed module
- decide if a space ($\geq$ NormedModule) is only zero
- norm on functions (`operator_norm`) on $\mathbb{R} \cup \{+\infty\}$

Summary of the work done

- results about functional spaces, linear and bilinear mappings
- fixed-point theorem in a sub-complete normed module
- decide if a space ($\geq$ NormedModule) is only zero
- norm on functions (`operator_norm`) on $\mathbb{R} \cup \{+\infty\}$
- 8 equivalences of continuity of linear mappings

Summary of the work done

- results about functional spaces, linear and bilinear mappings
- fixed-point theorem in a sub-complete normed module
- decide if a space ($\geq$ NormedModule) is only zero
- norm on functions (`operator_norm`) on $\mathbb{R} \cup \{+\infty\}$
- 8 equivalences of continuity of linear mappings
- definitions of pre-Hilbert and Hilbert spaces in Coquelicot hierarchy

Summary of the work done

- results about functional spaces, linear and bilinear mappings
- fixed-point theorem in a sub-complete normed module
- decide if a space ($\geq$ NormedModule) is only zero
- norm on functions (`operator_norm`) on $\mathbb{R} \cup \{+\infty\}$
- 8 equivalences of continuity of linear mappings
- definitions of pre-Hilbert and Hilbert spaces in Coquelicot hierarchy
- define `clm`: the set of the continuous linear mappings

Summary of the work done

- results about functional spaces, linear and bilinear mappings
- fixed-point theorem in a sub-complete normed module
- decide if a space ($\geq$ NormedModule) is only zero
- norm on functions (`operator_norm`) on $\mathbb{R} \cup \{+\infty\}$
- 8 equivalences of continuity of linear mappings
- definitions of pre-Hilbert and Hilbert spaces in Coquelicot hierarchy
- define `clm`: the set of the continuous linear mappings
- prove it is a `NormedModule`, to consider `clm` E (`clm` E $\mathbb{R}$)

Summary of the work done

- results about functional spaces, linear and bilinear mappings
- fixed-point theorem in a sub-complete normed module
- decide if a space ($\geq$ NormedModule) is only zero
- norm on functions (`operator_norm`) on $\mathbb{R} \cup \{+\infty\}$
- 8 equivalences of continuity of linear mappings
- definitions of pre-Hilbert and Hilbert spaces in Coquelicot hierarchy
- define `clm`: the set of the continuous linear mappings
- prove it is a `NormedModule`, to consider `clm` E (`clm` E $\mathbb{R}$)
- prove Lax-Milgram theorem and Céa's lemma

Lax-Milgram Theorem and Céa's Lemma

Theorem (Lax-Milgram)

Let E : `Hilbert`, $f \in E'$, $C, \alpha \in \mathbb{R}_+^*$. Let $\varphi : E \to$ `Prop`, φ `ModuleSpace`-compatible and complete. Let a be a bilinear form of E bounded by C and α-coercive. Then:

$$\exists! u \in E, \varphi(u) \wedge \forall v \in E, \varphi(v) \Longrightarrow f(v) = a(u, v) \wedge \|u\|_E \leq \frac{1}{\alpha} |||f|||_\varphi.$$

Lemma (Céa)

Let E : `Hilbert`, $f \in E'$, $0 < \alpha$. Let $\varphi : E \to$ `Prop`, φ `ModuleSpace`-compatible and complete. Let a be a bilinear form of E, bounded by $C > 0$ and α-coercive. Let u and u_φ be the solutions given by Lax–Milgram Theorem respectively on E and on the subspace φ. Then:

$$\forall v_\varphi \in E, \varphi(v_\varphi) \Longrightarrow \|u - u_\varphi\|_E \leq \frac{C}{\alpha} \|u - v_\varphi\|_E.$$

Outline

Lax-Milgram Theorem in constructive logic

The first version of this theorem was:

$\mathcal{H}_1 = (\texttt{phi} : \texttt{E} \to \texttt{Prop}) : \texttt{forall u} : \texttt{E}, \texttt{forall eps} : \texttt{posreal},$
$\quad \texttt{decidable} (\texttt{exists w} : \texttt{E}, \texttt{phi w} \wedge \texttt{norm (minus u w)} < \texttt{eps}).$

$\mathcal{H}_2 = (\texttt{phi} : \texttt{E} \to \texttt{Prop}) (\texttt{f} : \texttt{topo\_dual E}) :$
$\quad \texttt{decidable} (\texttt{exists u}, \neg \neg \texttt{phi u} \wedge \texttt{f u} \neq 0).$

Theorem (Lax-Milgram)

Let $f \in E'$, $0 < \alpha$. Let $\varphi : E \to \texttt{Prop}$, φ `ModuleSpace`-compatible and complete. Let a be a bilinear form on E, bounded and α-coercive.
Suppose $\forall f \in E', \ \mathcal{H}_1(\ker(f) \ \wedge \ \neg\neg\varphi) \ \wedge \ \mathcal{H}_2(\varphi, f)$.
Then, there exists a unique $u \in E$ such that $\neg\neg\varphi(u)$ and

$$\forall v \in E, \quad \neg\neg\varphi(v) \implies f(v) = a(u, v) \quad \wedge \quad \|u\|_E \leq \frac{1}{\alpha} \cdot |||f|||_{\varphi}.$$

Come to the dark side; we have axioms!

Given the previous experiment, we added the excluded middle, both for readability and for spreading formal methods to mathematicians.

Come to the dark side; we have axioms!

Given the previous experiment, we added the excluded middle, both for readability and for spreading formal methods to mathematicians.

For the rest of the talk, we assume:

Come to the dark side; we have axioms!

Given the previous experiment, we added the excluded middle, both for readability and for spreading formal methods to mathematicians.

For the rest of the talk, we assume:

- real number axiom(s) from the standard library,

Come to the dark side; we have axioms!

Given the previous experiment, we added the excluded middle, both for readability and for spreading formal methods to mathematicians.

For the rest of the talk, we assume:

- real number axiom(s) from the standard library,
- excluded middle,

Come to the dark side; we have axioms!

Given the previous experiment, we added the excluded middle, both for readability and for spreading formal methods to mathematicians.

For the rest of the talk, we assume:

- real number axiom(s) from the standard library,
- excluded middle,
- functional extensionality (a functional analysis must-have)

Come to the dark side; we have axioms!

Given the previous experiment, we added the excluded middle, both for readability and for spreading formal methods to mathematicians.

For the rest of the talk, we assume:

- real number axiom(s) from the standard library,
- excluded middle,
- functional extensionality (a functional analysis must-have)
- Hilbert's epsilon (I want the inverse of a bijective function)

Come to the dark side; we have axioms!

Given the previous experiment, we added the excluded middle, both for readability and for spreading formal methods to mathematicians.

For the rest of the talk, we assume:

- real number axiom(s) from the standard library,
- excluded middle,
- functional extensionality (a functional analysis must-have)
- Hilbert's epsilon (I want the inverse of a bijective function)

(All this may hurt you, but mathematicians do that all the time.)

Back to the FEM

Towards the Coq formalization of the finite element method:

- Lax-Milgram theorem (✓)

Back to the FEM

Towards the Coq formalization of the finite element method:

- Lax-Milgram theorem (✓)
- requires the subspace to be complete
  (✓ for finite-dimensional subspaces)

Back to the FEM

Towards the Coq formalization of the finite element method:

- Lax-Milgram theorem (✓)
- requires the subspace to be complete
  (✓ for finite-dimensional subspaces)
- requires E to be a Hilbert space

Back to the FEM

Towards the Coq formalization of the finite element method:

- Lax-Milgram theorem (✓)
- requires the subspace to be complete
  (✓ for finite-dimensional subspaces)
- requires E to be a Hilbert space
- E will be instantiated as the Sobolev space L_2

Back to the FEM

Towards the Coq formalization of the finite element method:

- Lax-Milgram theorem (✓)
- requires the subspace to be complete
  (✓ for finite-dimensional subspaces)
- requires E to be a Hilbert space
- E will be instantiated as the Sobolev space L_2

$\Rightarrow$ prove that L_2 is an Hilbert space

Back to the FEM

Towards the Coq formalization of the finite element method:

- Lax-Milgram theorem (✓)
- requires the subspace to be complete
  (✓ for finite-dimensional subspaces)
- requires E to be a Hilbert space
- E will be instantiated as the Sobolev space L_2

⇒ prove that L_2 is an Hilbert space

⇒ Lebesgue integration!

Back to the FEM

Towards the Coq formalization of the finite element method:

- Lax-Milgram theorem (✓)
- requires the subspace to be complete
  (✓ for finite-dimensional subspaces)
- requires E to be a Hilbert space
- E will be instantiated as the Sobolev space L_2

⇒ prove that L_2 is an Hilbert space

⇒ Lebesgue integration!

- . . .

Outline

Measurability

Given a set `E→ Prop`, is it measurable?

Measurability

Given a set `E→ Prop`, is it measurable?

We chose the definition from the generator sets:

```
Context {E : Type}.

(* initialization sets *)
Variable gen : (E → Prop) → Prop.

Inductive measurable : (E→ Prop) → Prop :=
  | measurable_gen : forall omega, gen omega → measurable omega
  | measurable_empty : measurable (fun _ ⇒ False)
  | measurable_compl : forall omega,
        measurable (fun x ⇒ not (omega x)) → measurable omega
  | measurable_union_countable :
   forall omega:nat → (E→ Prop),
     (forall n, measurable (omega n)) →
        measurable (fun x ⇒ exists n, omega n x).
```

The measurable sets are aka σ-algebras.

Measurability

Advantages of `Inductive`:
$\Rightarrow$ induction is possible
$\Rightarrow$ easy theorems

Measurability

Advantages of `Inductive`:
⇒ induction is possible
⇒ easy theorems

We defined generators on $\mathbb{R}$ and $\overline{\mathbb{R}}$:

```
Definition gen_R_cc := (fun om ⇒ exists a b, (forall x, om x ↔ a <= x <= b)).
Definition gen_Rbar_mc := (fun om ⇒ exists a, (forall x, om x ↔ Rbar_le a x)).
```

Measurability

Advantages of `Inductive`:
⇒ induction is possible
⇒ easy theorems

We defined generators on $\mathbb{R}$ and $\overline{\mathbb{R}}$:

```
Definition gen_R_cc := (fun om => exists a b, (forall x, om x <-> a <= x <= b)).
Definition gen_Rbar_mc := (fun om => exists a, (forall x, om x <-> Rbar_le a x)).
```

But we may use other generators and prove the measurable sets are the same. For instance `a` $<$ `x` $<$ `b` or with `a` and `b` in $\mathbb{Q}$.

Measurability

Advantages of `Inductive`:
⇒ induction is possible
⇒ easy theorems

We defined generators on $\mathbb{R}$ and $\overline{\mathbb{R}}$:

```
Definition gen_R_cc := (fun om ⇒ exists a b, (forall x, om x ↔ a <= x <= b)).
Definition gen_Rbar_mc := (fun om ⇒ exists a, (forall x, om x ↔ Rbar_le a x)).
```

But we may use other generators and prove the measurable sets are the same. For instance `a < x < b` or with `a` and `b` in $\mathbb{Q}$.

And we proved that it is equivalent to the usual Borel σ-algebras:

```
Lemma measurable_R_open : forall om,
  measurable gen_R_cc om ↔ measurable open om.
```

Measurable functions

A function $f : E \to F$ is measurable if the set $A(f(x))$ is measurable in F for all measurable sets A in E:

```
Definition measurable_fun : (E → F) → Prop :=
  fun f ⇒ (forall (A: F → Prop), measurable genF A →
  measurable genE (fun x ⇒ A (f x))).
```

Measurable functions

A function $f : E \to F$ is measurable if the set $A(f(x))$ is measurable in F for all measurable sets A in E:

```
Definition measurable_fun : (E → F) → Prop :=
  fun f ⇒ (forall (A: F → Prop), measurable genF A →
  measurable genE (fun x ⇒ A (f x))).
```

The sum and multiplication by a scalar of measurable functions on $\mathbb{R}$ and $\overline{\mathbb{R}}$ are measurable functions.

Measure definition

We choose to not (yet) define the Lebesgue measure, but define what a measure is supposed to satisfy:

```
Context {E : Type}.
Variable gen : (E → Prop) → Prop.

Record measure := mk_measure {
   meas :> (E→ Prop) → Rbar ;
   meas_False : meas (fun _ ⇒ False) = 0 ;
   meas_ge_0: forall om, Rbar_le 0 (meas om) ;
   meas_sigma_additivity : forall omega :nat → (E→ Prop),
     (forall n, measurable gen (omega n)) →
     (forall n m x, omega n x → omega m x → n = m)
     → meas (fun x ⇒ exists n, omega n x)
             = Sup_seq (fun n ⇒ sum_Rbar n (fun m ⇒ meas (omega m)))
}.
```

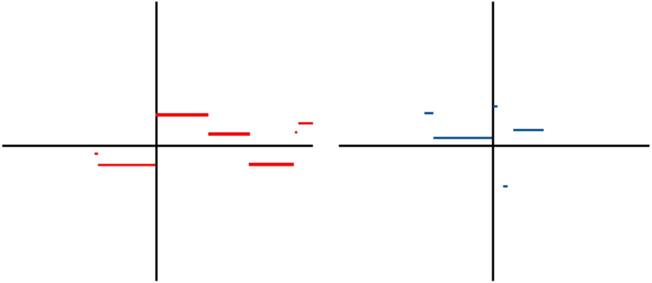
Measure properties

Many properties hold for all measures such as:

```
Lemma measure_Boole_ineq : forall (mu:measure) (A:nat → E→ Prop) (N : nat),
    (forall n, n <= N → measurable gen (A n)) →
      Rbar_le (mu (fun x ⇒ exists n, n <= N ∧ A n x))
              (sum_Rbar N (fun m ⇒ mu (A m))).
```

$$\mu\left(\bigcup_{i\in[0..N]} A_i\right) \leq \sum_{i\in[0..N]} \mu(A_i)$$

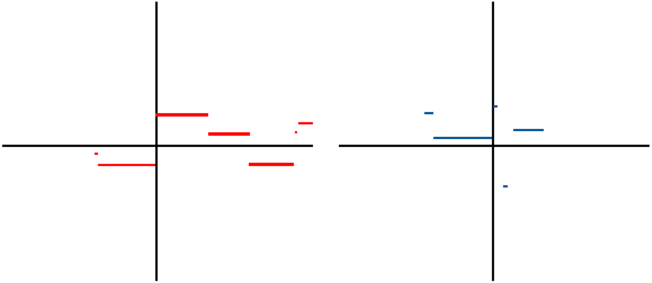
Simple functions?



Examples of simple functions @ mathonline

$$f = \sum_{y \in f(E)} \mathbb{1}_{f^{-1}(\{y\})}$$

Simple functions?



Examples of simple functions @ mathonline

$$f = \sum_{y \in f(E)} \mathbb{1}_{f^{-1}(\{y\})}$$

We have tried various definitions of simple functions, especially as we prefer to sum over a finite set of values.

Simple functions definition

```
Definition finite_vals : (E→ R) → (list R) → Prop :=
    fun f l ⇒ forall y, In (f y) l.
```

⇒ OK, but not unique.

Simple functions definition

```
Definition finite_vals : (E→ R) → (list R) → Prop :=
    fun f l ⇒ forall y, In (f y) l.
```

⇒ OK, but not unique.

```
Definition finite_vals_canonic : (E→ R) → (list R) → Prop :=
  fun f l ⇒ (LocallySorted Rlt l) ∧
             (forall x, In x l → exists y, f y = x) ∧
             (forall y, In (f y) l).
```

⇒ unique!
We were able to construct the second list from the first.

Simple functions integration

$$\int f \, d\mu \overset{\text{def.}}{=} \sum_{a \in f(X)} a \, \mu\left(f^{-1}(a)\right) \quad \in \overline{\mathbb{R}}$$

Simple functions integration

$$\int f \, d\mu \overset{\text{def.}}{=} \sum_{a \in f(X)} a \, \mu\left(f^{-1}(a)\right) \quad \in \overline{\mathbb{R}}$$

```
Definition SF_aux : (E→ R) → (list R) → Prop :=
  fun f l ⇒ finite_vals_canonic f l ∧
   (forall a, measurable gen (fun x ⇒ f x = a)).

Definition SF : (E→ R) → Set := fun f ⇒ { l | SF_aux f l}.

Definition af1 (f:E→ R) :=
  (fun a :  Rbar ⇒ Rbar_mult a (mu (fun (x:E) ⇒ f x = a))).

Definition LInt_simple_fun_p :=
   fun (f:E→ R) (H:SF gen f) ⇒ let l:= (proj1_sig H) in
     sum_Rbar_map l (af1 f).
```

We proved the value does not depends on the proof `H`.

Simple functions integration

$$\int f \, d\mu \stackrel{\text{def.}}{=} \sum_{a \in f(X)} a \, \mu\left(f^{-1}(a)\right) \quad \in \overline{\mathbb{R}}$$

```
Definition SF_aux : (E→ R) → (list R) → Prop :=
  fun f l ⇒ finite_vals_canonic f l ∧
   (forall a, measurable gen (fun x ⇒ f x = a)).

Definition SF : (E→ R) → Set := fun f ⇒ { l | SF_aux f l}.

Definition af1 (f:E→ R) :=
  (fun a : Rbar ⇒ Rbar_mult a (mu (fun (x:E) ⇒ f x = a))).

Definition LInt_simple_fun_p :=
   fun (f:E→ R) (H:SF gen f) ⇒ let l:= (proj1_sig H) in
     sum_Rbar_map l (af1 f).
```

We proved the value does not depends on the proof `H`.

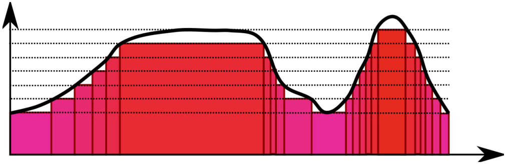
⇒ theorems about sum, multiplication by a scalar and change of variable

Lebesgue integral

$$\int f \, d\mu \stackrel{\text{def.}}{=} \sup_{\varphi \in \mathcal{SF}_+, \varphi \leq f} \int \varphi \, d\mu \quad \in \overline{\mathbb{R}}$$

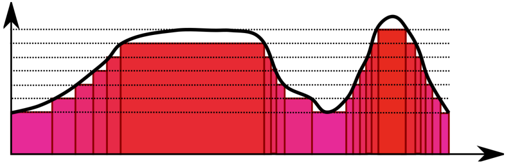
Lebesgue integral

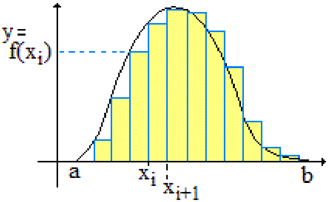
$$\int f \, d\mu \overset{\text{def.}}{=} \sup_{\varphi \in \mathcal{SF}_+, \varphi \leq f} \int \varphi \, d\mu \quad \in \overline{\mathbb{R}}$$



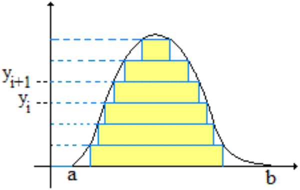
Lebesgue integral

$$\int f \, d\mu \overset{\text{def.}}{=} \sup_{\varphi \in \mathcal{SF}_+, \varphi \leq f} \int \varphi \, d\mu \quad \in \overline{\mathbb{R}}$$





Riemann integral *vs* Lebesgue integral



© Serge Mehl chronomath.com

Lebesgue integral definition

$$\int f \, d\mu \stackrel{\text{def.}}{=} \sup_{\varphi \in \mathcal{SF}_+, \varphi \leq f} \int \varphi \, d\mu \quad \in \overline{\mathbb{R}}$$

```
Definition LInt_p :(E→ Rbar) → Rbar := fun f ⇒
     Rbar_lub (fun x ⇒ exists (g:E→ R), exists (Hg: SF gen g),
                      non_neg g ∧
                      (forall (z:E), Rbar_le (g z) (f z)) ∧
                      LInt_simple_fun_p mu g Hg = x).
```

A monotone convergence theorem

Theorem (Beppo Levi, monotone convergence)

Let $(f_n)_{n\in\mathbb{N}} \in \mathcal{M}_+$ be a sequence of nonnegative measurable functions, that is pointwise nondecreasing. Then, the pointwise limit of $(f_n)_{n\in\mathbb{N}}$ is nonnegative and measurable, and we have in $\overline{\mathbb{R}}$

$$\int \lim_{n\to\infty} f_n \, d\mu = \lim_{n\to\infty} \int f_n \, d\mu.$$

Note that $\lim_{n\to\infty} f_n = \sup_{n\in\mathbb{N}} f_n$ and $\lim_{n\to\infty} \int f_n = \sup_{n\in\mathbb{N}} \int f_n$.

```
Lemma Beppo_Levi : ∀ f : nat → E → Rbar,
  (∀ n, non_neg (f n)) → (∀ n, measurable_fun_Rbar genE (f n)) →
  (∀ x n, Rbar_le (f n x) (f (S n) x)) →
  LInt_p μ (fun x ⇒ Sup_seq (fun n ⇒ f n x)) = Sup_seq (fun n ⇒ LInt_p μ (f n)).
```

Focus on a hard theorem

Focus on a hard theorem

$$\int (f+g) = \int f + \int g$$

```
Lemma LInt_p_plus : forall f g,
    non_neg f → non_neg g →
    measurable_fun_Rbar gen f → measurable_fun_Rbar gen g →
    LInt_p mu (fun x ⇒ Rbar_plus (f x) (g x))
        = Rbar_plus (LInt_p mu f) (LInt_p mu g).
```

Proof of $\int(f+g) = \int f + \int g$ (1/2)

It needs adapted sequences:

```
Definition is_adapted_seq (f:E→ Rbar) (phi:nat→ E→ R) :=
    (forall n, non_neg (phi n)) ∧
    (forall (x:E) n,  phi n x <= phi (S n) x) ∧
    (forall n, exists l, SF_aux gen (phi n) l) ∧
    (forall (x:E),  is_sup_seq (fun n ⇒ phi n x) (f x)).
```

as their limit gives the integral:

```
Lemma LInt_p_with_adapted_seq :
   forall f phi, is_adapted_seq f phi →
      is_sup_seq (fun n ⇒ LInt_p mu (phi n)) (LInt_p mu f).
```

Proof of $\int(f+g) = \int f + \int g$ (2/2)

Adapted sequences may be defined like that:

$$\forall x, \quad f_n(x) \stackrel{\text{def.}}{=} \begin{cases} \dfrac{\lfloor 2^n f(x) \rfloor}{2^n} & \text{when } f(x) < n, \\ n & \text{otherwise.} \end{cases}$$

Proof of $\int(f+g) = \int f + \int g$ (2/2)

Adapted sequences may be defined like that:

$$\forall x, \quad f_n(x) \stackrel{\text{def.}}{=} \begin{cases} \dfrac{\lfloor 2^n f(x) \rfloor}{2^n} & \text{when } f(x) < n, \\ n & \text{otherwise.} \end{cases}$$

that may be written in Coq as:

```
Definition mk_adapted_seq (n:nat) (x:E) :=
  match (Rbar_le_lt_dec (INR n) (f x)) with
           | left _ ⇒ INR n
           | right _ ⇒ round radix2 (FIX_exp (−n)) Zfloor (f x)
  end.
```

relying on fixed-point arithmetic defined by the Flocq library!!

And then:

```
Lemma mk_adapted_seq_is_adapted_seq :
   is_adapted_seq f mk_adapted_seq.
```

Outline

Definition of a finite element (geometry and properties)

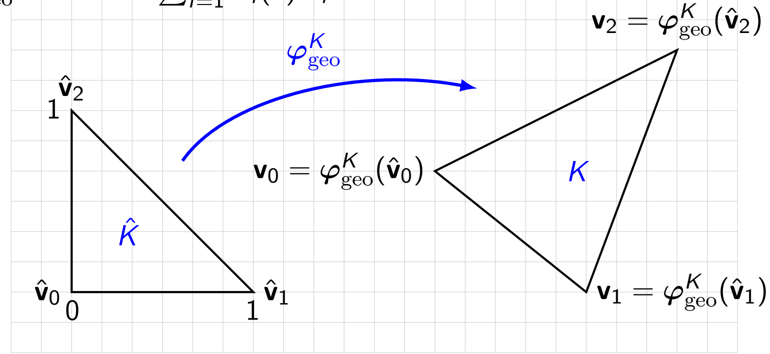
Definition of a finite element (geometry and properties)

```
Record FE : Type := mk_FE {
  d : nat ; (* space dimension eg 1, 2, 3 *)
  ndof : nat ; (* nb of degrees of freedom - eg number of nodes for Lagrange.
  d_pos :    (0 < d)%coq_nat ;
  ndof_pos : (0 < ndof)%coq_nat ;
  g_family : geom_family ; (* either Simplex or Quad *)
  nvtx : nat := (* ... *) (* number of vertices *)
  vtx : '(' R^d)^nvtx ; (* vertices of geometrical element *)
  K_geom : 'R^d → Prop := convex_envelop vtx ; (* geometrical element *)
  P_approx : FRd d → Prop ; (* Subspace of F *)
  P_compat_fin : has_dim P_approx ndof ;
  sigma : '( FRd d → R)^ndof ;
  is_linear_mapping_sigma : forall i, is_linear_mapping (sigma i) ;
  FE _is_unisolvent :
       is_unisolvent d ndof P_approx P_compat_fin (gather sigma) ;
 }.
```

Geometrical transformation

Goal: to transform (and back) a finite element into a reference (nice) finite element.

$\varphi^K_{\mathrm{geo}} : \hat{K} \ni \hat{\mathbf{x}} \mapsto \sum_{i=1}^{n_V^K} \hat{\mathcal{L}}_i(\hat{\mathbf{x}})\, \mathbf{v}_i$ affine



Lagrange nodes

We want to define the simplest finite elements (Lagrange finite elements).

There are still admits left.

Lagrange nodes

We want to define the simplest finite elements (Lagrange finite elements).

There are still admits left.

Let us focus on one point (one with no admit left :)

Given d and k, I want the list of the vectors of $\mathbb{N}^d$ such that their sum is smaller than k (Lagrange nodes).

Lagrange nodes

We want to define the simplest finite elements (Lagrange finite elements).

There are still admits left.

Let us focus on one point (one with no admit left :)

Given d and k, I want the list of the vectors of $\mathbb{N}^d$ such that their sum is smaller than k (Lagrange nodes).

- is this combinatorics?

Lagrange nodes

We want to define the simplest finite elements (Lagrange finite elements).

There are still admits left.

Let us focus on one point (one with no admit left :)

Given d and k, I want the list of the vectors of $\mathbb{N}^d$ such that their sum is smaller than k (Lagrange nodes).

- is this combinatorics?
- or geometry? (see later)

Lagrange nodes

We want to define the simplest finite elements (Lagrange finite elements).

There are still admits left.

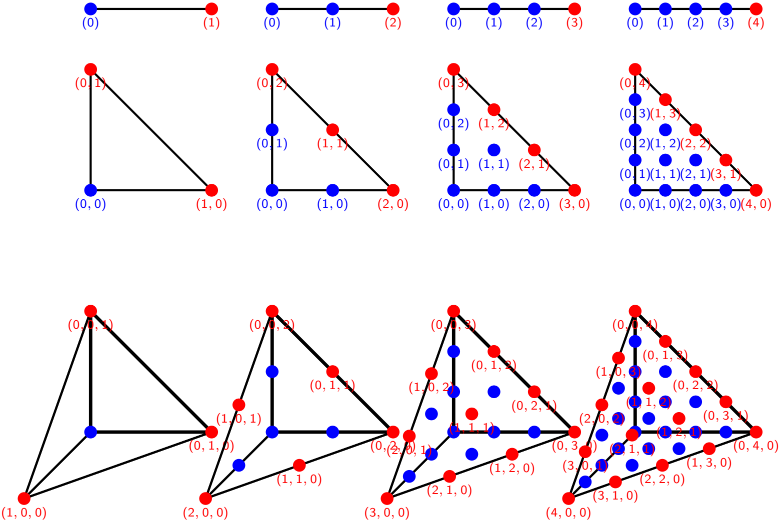
Let us focus on one point (one with no admit left :)

Given d and k, I want the list of the vectors of $\mathbb{N}^d$ such that their sum is smaller than k (Lagrange nodes).

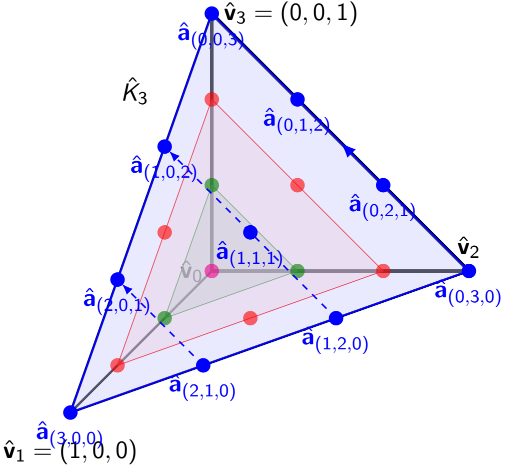
- is this combinatorics?
- or geometry? (see later)
- or analysis?

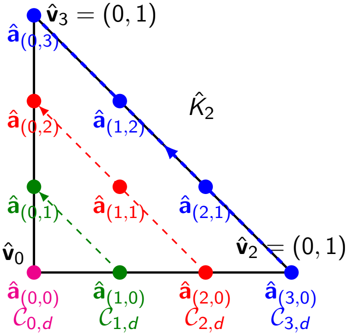
  Such a vector can be seen as a monomial among the polynomials on d variables of degree $\leq k$.

For $d = 1, 2, 3$ (in line) and $k = 1, 2, 3, 4$ (in column).



Construction of Lagrange finite element nodes





Coq formalization

We defined this list of vectors $\mathcal{A}_d^k$ (of size the binomial coefficient C_{d+k}^d) by concatenation of lists of varying sizes.

```
Lemma A_d_k_sum : forall d k i,
    (sum (A_d_k d k i) <= k)%coq_nat.

Lemma A_d_k_surj : forall d k (b:'nat^d),
    (sum b <= k)%coq_nat → exists i, b = A_d_k d k i.

Lemma A_d_k_inj : forall d k, injective (A_d_k d k).

Lemma A_d_k_MOn : forall d k, is_orderedF MOn (A_d_k d k).
    (* a special order near the grevlex order on monomials *)
```

Outline

Conclusion

A journey

- hand by hand with mathematicians,
- pretty long,
- with various mathematics inside.

Available at https://lipn.univ-paris13.fr/coq-num-analysis/
and as an opam package.

Conclusion

A journey

- hand by hand with mathematicians,
- pretty long,
- with various mathematics inside.

Available at https://lipn.univ-paris13.fr/coq-num-analysis/ and as an opam package.

Difficult parts:

- handling subspaces,
- trade-off between a usable library and proving one main theorem.

Perspectives

- end the definition of Lagrange finite elements

Perspectives

- end the definition of Lagrange finite elements
- write the corresponding article about the FEM

Perspectives

- end the definition of Lagrange finite elements
- write the corresponding article about the FEM
- extend Lebesgue integrals to functions of varying sign

$$\int f = \int \max(0, f) - \int \max(0, -f)$$

(or by Bochner integral)

Perspectives

- end the definition of Lagrange finite elements
- write the corresponding article about the FEM
- extend Lebesgue integrals to functions of varying sign

$$\int f = \int \max(0, f) - \int \max(0, -f)$$

  (or by Bochner integral)
- define L_2 and prove it is a Hilbert

Perspectives

- end the definition of Lagrange finite elements
- write the corresponding article about the FEM
- extend Lebesgue integrals to functions of varying sign

$$\int f = \int \max(0, f) - \int \max(0, -f)$$

  (or by Bochner integral)
- define L_2 and prove it is a Hilbert
- define the FEM algorithm and prove it

Perspectives

- end the definition of Lagrange finite elements
- write the corresponding article about the FEM
- extend Lebesgue integrals to functions of varying sign

$$\int f = \int \max(0, f) - \int \max(0, -f)$$

  (or by Bochner integral)
- define L_2 and prove it is a Hilbert
- define the FEM algorithm and prove it
- prove a real implementation (in floating-point arithmetic)

Thank you for your attention