EuroProofNet Workshop on Proof Libraries
Orsay

Numerical Analysis in Rocq
Simplicial Lagrange Finite Elements

Sylvie Boldo,
François Clément, Vincent Martin, Micaela Mayero, Houda Mouhcine

Inria, Université Paris-Saclay

September 16th, 2025

Outline

Motivations

𝔐𝔞𝔱𝔥𝔢𝔪𝔞𝔱𝔦𝔠𝔰

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

Motivations

𝔐𝔞𝔱𝔥𝔢𝔪𝔞𝔱𝔦𝔠𝔰

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

Applied Mathematics

numerical scheme, convergence
algorithms + theorems

Motivations

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

Motivations

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

Applied M… — …me, convergence

Computer Science — float… …entation
programs + …

Formal Proof

Motivations

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

Motivations

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

Too complex to be solved by an exact formula
$\Rightarrow$ approximated by numerical schemes on meshes

$\Rightarrow$ mathematical proof of the convergence of the numerical scheme
(one computes values nearer to the solution when the mesh size decreases)

Motivations

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

Too complex to be solved by an exact formula
$\Rightarrow$ approximated by numerical schemes on meshes

$\Rightarrow$ mathematical proof of the convergence of the numerical scheme
(one computes values nearer to the solution when the mesh size decreases)

$\Rightarrow$ real program that implements this scheme or this method

Motivations

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

Too complex to be solved by an exact formula
$\Rightarrow$ approximated by numerical schemes on meshes

$\Rightarrow$ mathematical proof of the convergence of the numerical scheme
(one computes values nearer to the solution when the mesh size decreases)

$\Rightarrow$ real program that implements this scheme or this method

$\Rightarrow$ **towards verifying these programs!**

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...

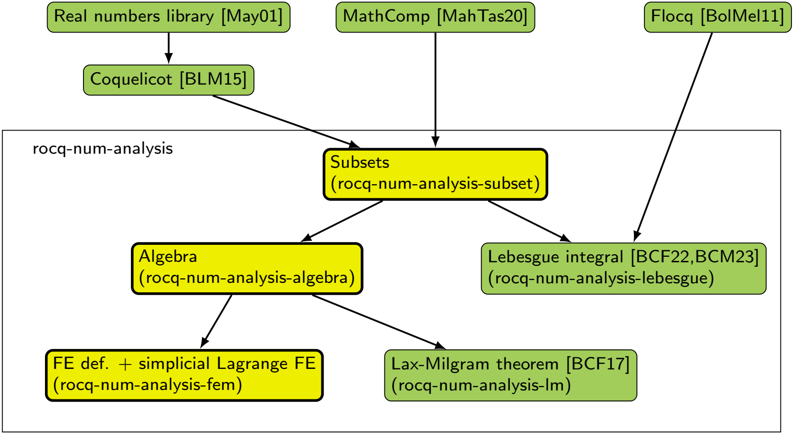
The conclusion of the investigation was that the loss was caused by a failure in a cell wall, resulting in a serious crack and a leakage that the pumps were not able to cope with. The wall failed as a result of a combination of a serious error in the finite element analysis and insufficient anchorage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

Outline

Outline

State of the lib

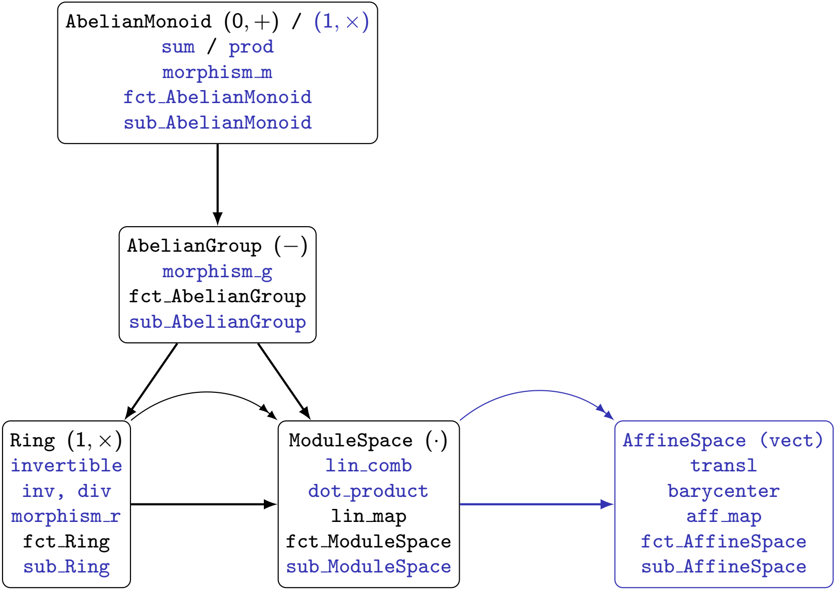


(155 files, 85 kLoC)

Subsets and Algebra

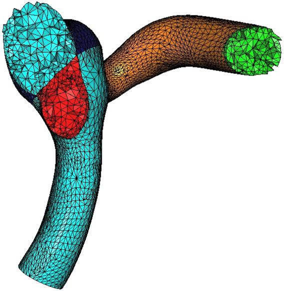
- Subsets
  - subsets as `E`$\rightarrow$ `Prop`,
  - function restrictions and their properties.
- Algebra
  - finite families (a.k.a vectors),
  - linear and affine algebra,
  - a hierarchy of algebraic structures (see next slide) built upon Coquelicot.

Hierarchy of algebraic structures (incl. Coquelicot in black)



Outline

Mesh



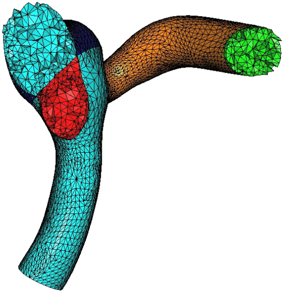
@ V. Martin

On a complex geometry, one has a mesh, made of triangles / tetrahedrons.

The Finite Element Method aims to approximate the PDE on each element, while constructing a continuous solution.

A Finite Element is a geometric element + ??? to solve the PDE on it.

Mesh



@ V. Martin

On a complex geometry, one has a mesh, made of triangles / tetrahedrons.

The Finite Element Method aims to approximate the PDE on each element, while constructing a continuous solution.

A Finite Element is a geometric element + ??? to solve the PDE on it.

A **Finite Element** is how to approximate a function (solution of the PDE) by a simpler function (eg. polynomial) on a small simple geometric cell (eg. tetrahedron), by retaining particular values (eg. point values, mean values, fluxes through interfaces) to reduce to a finite-dimension problem.

Definition of a Finite Element

A Finite Element is a triple $(\mathcal{K}, \mathcal{P}, \Sigma)$:

1. $\mathcal{K}$ is a geometric cell, such as a simplex (segment in $\mathbb{R}$, triangle in $\mathbb{R}^2$, tetrahedron in $\mathbb{R}^3$)
2. $\mathcal{P}$ is a vector space of functions on $\mathcal{K}$, of dimension n_{dof} (often polynomials with bounded degree)
3. Σ is composed of n_{dof} linear forms on $\mathcal{P}$.

A Finite Element must have the unisolvence property:
For $\Sigma := \{\sigma_i\}_{i \in \{0:n_{dof}-1\}}$, let $\Phi_\Sigma : \mathcal{P} \to \mathbb{R}^{n_{dof}}$ be
$\Phi_\Sigma(p) := (\sigma_i(p))_{i \in \{0:n_{dof}-1\}}$. Unisolvence means that Φ_Σ is a bijection.

Corresponding Rocq Definition

A FE is either a simplex or a cuboid with the correct number of vertices:

```
Inductive shape_type := Simplex | Cuboid.

Definition nvtx_of_shape (d : ℕ) (shp : shape_type) : ℕ :=
  match shp with Simplex ⇒ d.+1 | Cuboid ⇒ 2^d end.
```

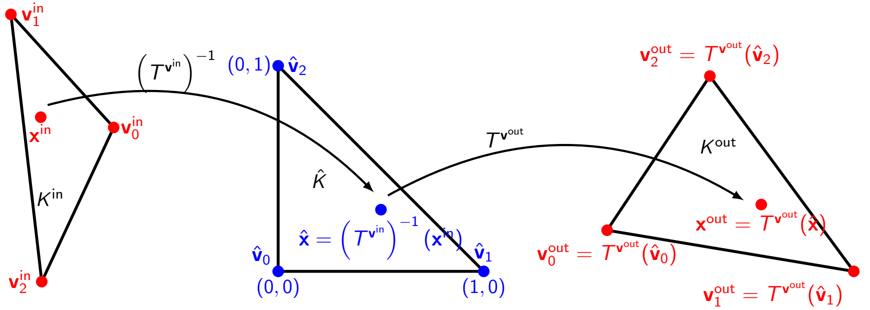
Corresponding Rocq Definition

A FE is either a simplex or a cuboid with the correct number of vertices:

```
Inductive shape_type := Simplex | Cuboid.

Definition nvtx_of_shape (d : ℕ) (shp : shape_type) : ℕ :=
  match shp with Simplex ⇒ d.+1 | Cuboid ⇒ 2^d end.

Record FE (d : ℕ) : Type := mk_FE {
  shape : shape_type;
  nvtx : ℕ := nvtx_of_shape d shape;
  K_vertices : [0..nvtx) → 'ℝ^d;
  K_geom : 'ℝ^d → Prop := convex_hull K_vertices; (* geometric element K *)
  ndof : ℕ;
  P_approx : FRd d → Prop;                        (* approximation space P *)
  P_approx_has_dim : has_dim P_approx ndof;
  S_dof : [0..ndof) → FRd d → ℝ;                   (* degrees of freedom Σ *)
  S_dof_lm : ∀ i, lin_map (S_dof i);
  unisolvence_inj : KerS0 P_approx (gather S_dof);
}.
```

Outline

Geometric Transformation of a Finite Element



From an input finite element and the output vertices $\mathbf{v}^{\text{out}}$, we can build an output finite element, relying on the geometric transformation $T_{\mathbf{v}}$ (based on Lagrange polynomials).

Instantiation!

Let us define from scratch a FE to check there is no inconsistency.

We build a d-FE depending on a variable $k \in \mathbb{N}$. So we need:

Instantiation!

Let us define from scratch a FE to check there is no inconsistency.

We build a d-FE depending on a variable $k \in \mathbb{N}$. So we need:

- a shape: `Simplex`, ✓

Instantiation!

Let us define from scratch a FE to check there is no inconsistency.

We build a d-FE depending on a variable $k \in \mathbb{N}$. So we need:

- a shape: `Simplex`, ✓
- vertices: the right triangle/tetrahedron/... of dimension d with sides of length 1, ✓

Instantiation!

Let us define from scratch a FE to check there is no inconsistency.

We build a d-FE depending on a variable $k \in \mathbb{N}$. So we need:

- a shape: `Simplex`, ✓
- vertices: the right triangle/tetrahedron/... of dimension d with sides of length 1, ✓
- `P_approx` is the vector space of polynomials from $\mathbb{R}^d$ to $\mathbb{R}$ of total degree $\leq k$, ✓

Instantiation!

Let us define from scratch a FE to check there is no inconsistency.

We build a d-FE depending on a variable $k \in \mathbb{N}$. So we need:

- a shape: `Simplex`, ✓
- vertices: the right triangle/tetrahedron/... of dimension d with sides of length 1, ✓
- `P_approx` is the vector space of polynomials from $\mathbb{R}^d$ to $\mathbb{R}$ of total degree $\leq k$, ✓
- `S_dof` are the linear forms that take a function and evaluate it at given points for evenly distributed points (called Lagrange nodes), ✓

Instantiation!

Let us define from scratch a FE to check there is no inconsistency.

We build a d-FE depending on a variable $k \in \mathbb{N}$. So we need:

- a shape: `Simplex`, ✓
- vertices: the right triangle/tetrahedron/... of dimension d with sides of length 1, ✓
- `P_approx` is the vector space of polynomials from $\mathbb{R}^d$ to $\mathbb{R}$ of total degree $\leq k$, ✓
- `S_dof` are the linear forms that take a function and evaluate it at given points for evenly distributed points (called Lagrange nodes), ✓
- a unisolvence proof (Φ_Σ is a bijection). ✓

$\mathcal{A}_d^k$

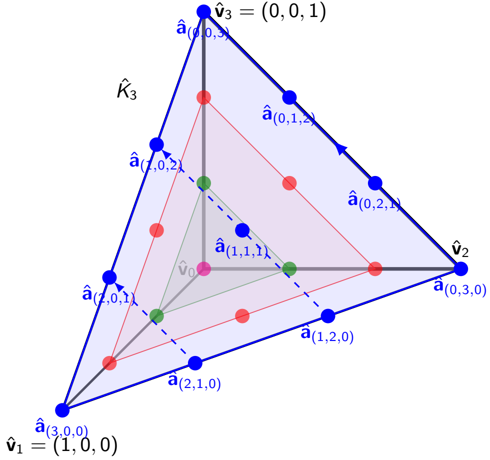
Given d and k, I want the family of families of $\mathbb{N}^d$ with a sum $\leq k$.

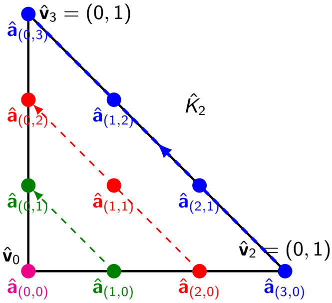
$\hookrightarrow$ useful in geometry for Lagrange nodes,
$\hookrightarrow$ useful for the monomials on $\mathbb{R}^d$ of degree $\leq k$: $(\mathbf{X}^\alpha)_{\alpha \in \mathcal{A}_d^k}$.

We define $\mathcal{A}_d^k$ (of size $\binom{d+k}{d}$) by concatenation and induction.

```
Lemma Adk_sum : forall d k idk, (sum (Adk d k idk) ≤ k).

Lemma Adk_surj : forall d k (b : ℕ^d), (sum b ≤ k) → { idk | b = Adk d k idk }.

Lemma Adk_inj : forall d k, injective (Adk d k).

Lemma Adk_sortedF : forall d k, sortedF grsymlex_lt (Adk d k).
    (* a special order near the grevlex order on monomials *)
```

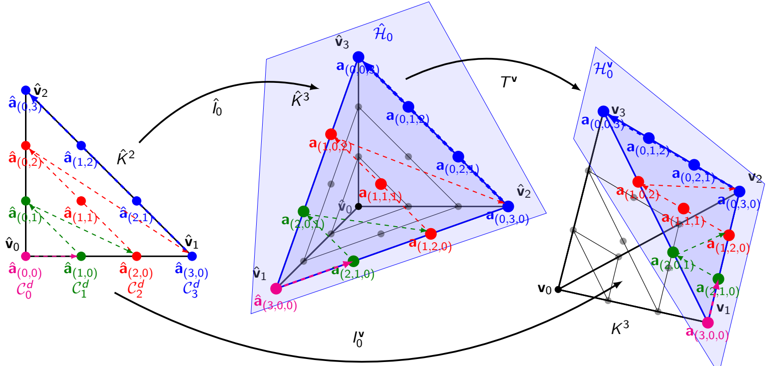
Reference tetrahedron and triangle with Lagrange nodes





Unisolvence

- long and hard proof,
- begins with a double induction on d and k,
- needs factorization of polynomials,
- needs injection onto a face hyperplane:



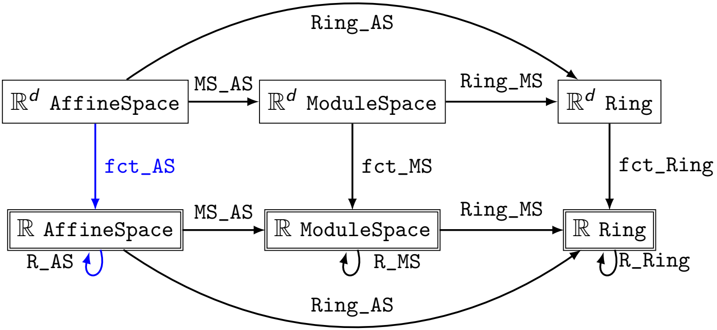
Outline

Difficulties

- chosen proof path: less analysis, and more affine arguments,
- unisolvence proof,
- substructures/restrictions: `E` $\rightarrow$ `Prop` is not a `ModuleSpace`,

Difficulties

- chosen proof path: less analysis, and more affine arguments,
- unisolvence proof,
- substructures/restrictions: `E` $\rightarrow$ `Prop` is not a `ModuleSpace`,
- canonical structures. For instance, $\mathbb{R}^d$ is an affine space because:



Conclusion

More details:

- opam package(s): `rocq-num-analysis`
- git repo: `https://lipn.univ-paris13.fr/rocq-num-analysis/`
- paper about FE: `https://inria.hal.science/hal-05128954`

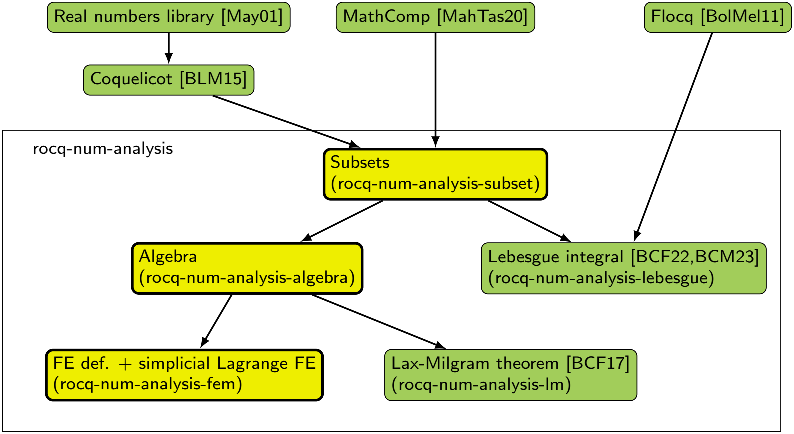
Conclusion

More details:

- opam package(s): `rocq-num-analysis`
- git repo: `https://lipn.univ-paris13.fr/rocq-num-analysis/`
- paper about FE: `https://inria.hal.science/hal-05128954`

- interdisciplinary work: mathematics / computer science,
- long-term work.

Perspectives



Perspectives