

# Enveloppes convexes et cartes combinatoires en Coq

---

Christophe Brun, Jean-François Dufourd, Nicolas Magaud



Journées Informatique et Géométrie 2007  
INRIA Sophia-Antipolis  
14-15 Juin 2007

# Objectif et contexte

---

Améliorer les techniques de programmation  
en modélisation et en algorithmique géométrique

Moyens utilisés :

- Spécifications formelles et preuves de programmes
- Modélisation géométrique à base topologique :  
représentations par les bords

Concrètement à Strasbourg :

- Calcul des constructions inductives et preuves interactives en Coq
- Modèles de cartes combinatoires

# Travaux connexes

---

- Y. Bertot and P. Castéran. *Interactive theorem proving and program development - Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.
- J.-F. Dufourd. Design and Formal Proof of a New Optimal Image Segmentation Program with Hypermaps. *Pattern Recognition*, 2007.
- Donald Knuth. *Axioms and Hulls*. LNCS, Springer-Verlag, 1991.
- L. I. Meikle and J. D. Fleuriot. Mechanical Theorem Proving in Computational Geometry. In *Automated Deduction in Geometry 2004*, Springer-Verlag, 2005.
- D. Pichardie and Y. Bertot. Formalizing Convex Hull Algorithms. In *Theorem Proving in Higher Order Logics (TPHOLs'01)*, Springer-Verlag, 2001.

# Dans cet exposé : un exemple concret

---

## Le calcul de l'enveloppe convexe d'un ensemble de points du plan

1. Modélisation du problème avec des cartes combinatoires
2. Description d'un algorithme (l'algorithme incrémental) en Coq
3. Extraction vers Ocaml et prototypage  
(saisie des données / visualisation du résultat)
4. Preuve de correction avec Coq : aspects topologiques / aspects géométriques

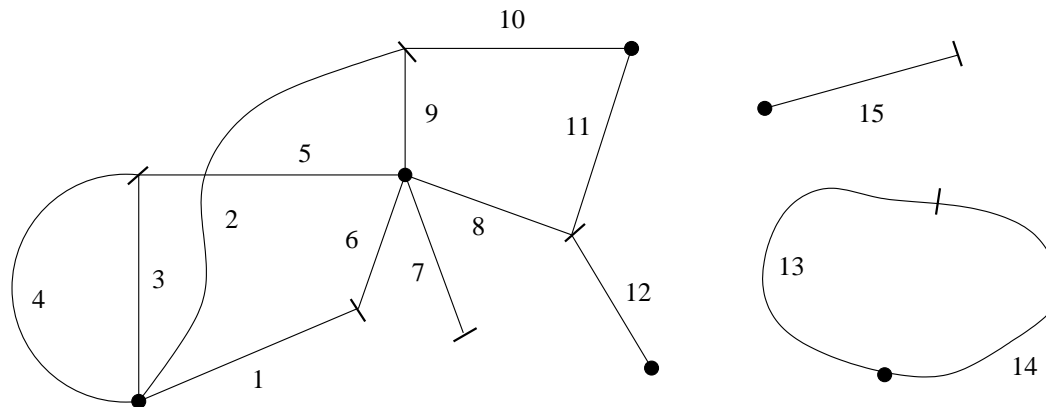
# 1. Hypercartes

---

## Définition 1 (Hypercarte)

- (i) Une hypercarte est une structure algébrique  $M = (D, \alpha_0, \alpha_1)$ , où  $D$  est un ensemble *fini* de brins, et  $\alpha_0, \alpha_1$  sont des permutations dans  $D$ .
- (ii) si  $y = \alpha_k(x)$ ,  $y$  est le  $k$ -successeur de  $x$ ,  $x$  est le  $k$ -prédécesseur de  $y$ , et  $x$  et  $y$  sont dits  $k$ -liés, ou  $k$ -cousus.

D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\alpha_0$	6	9	5	3	4	1	7	12	10	2	8	11	14	13	15
$\alpha_1$	2	3	4	1	6	7	8	9	5	11	10	12	14	13	15



# 1. Hypercartes

---

**Définition 2** (Orbites et cellules d'hypercarte)

(i) Soit  $D$  un ensemble et  $f_1, \dots, f_n$  des fonctions dans  $D$ . L'orbite de  $x \in D$  pour ces fonctions est le sous-ensemble de  $D$  noté

$\langle f_1, \dots, f_n \rangle (x)$  des éléments accessibles depuis  $x$  par n'importe quelle composition des fonctions  $f_1, \dots, f_n$ .

(ii) Dans l'hypercarte  $M = (D, \alpha_0, \alpha_1)$ ,  $\langle \alpha_0 \rangle (x)$  est la 0-orbite ou l'*arête* du brin  $x$ ,  $\langle \alpha_1 \rangle (x)$  sa 1-orbite ou son *sommet*,

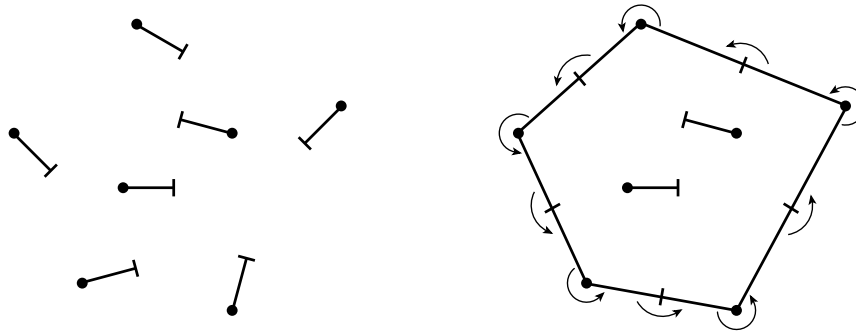
$\langle \alpha_1^{-1} \circ \alpha_0^{-1} \rangle (x)$  sa *face* et  $\langle \alpha_0, \alpha_1 \rangle (x)$  sa *composante connexe*.

Sommets, arêtes et faces sont aussi appelés *k-cellules topologiques* pour  $k = 0, 1$  et  $2$ .

## 2. Structures de données et modélisation

---

- Décidabilité pour les propriétés géométriques et aspects numériques
  - Hypothèse initiale : les points sont en position générale
  - Axiomatique de Knuth pour le prédicat d'orientation **ccw**  
Un triplet  $ccw(p, q, r)$  est toujours orienté soit dans le sens des aiguilles d'une montre, soit dans le sens inverse.
- Représentation des données initiales et du résultat



- Programmation par récursion structurelle en Coq

## 2.1. Représentation concrète des cartes

---

- Cartes libres

Inductive fmap : Set :=

  V : fmap

  | I : fmap -> dart -> point -> fmap

  | L : fmap -> dim -> dart -> dart -> fmap.

exemple : L (I (I 1 (x1,y1) V) 2 (x2,y2)) di1 1 2

- Préconditions pour obtenir des quasi-hypercartes

Definition prec\_I (m:fmap)(x:dart) : Prop :=

  x <> nil /\ ~ exd m x .

Definition prec\_Lq (m:fmap)(k:dim)(x y:dart) : Prop :=

  exd m x /\ exd m y /\ ~ succ m k x /\ ~ pred m k y.



## 2.1. Propriétés formelles sur les cartes

---

- **Aucun axiome** - définitions inductives simples
- Nombreux résultats significatifs obtenus  
**Relation d'Euler, Théorème du genre et Critère de planarité**

**Définition** (Caractéristique d'Euler, genre, planarité)

- (i) La caractéristique d'Euler de  $M$  est  $\chi = v + e + f - d$ .
- (ii) Le genre de  $M$  est  $g = c - \chi/2$ .
- (iii) Quand  $g = 0$ , l'hypercarte est dite planaire.

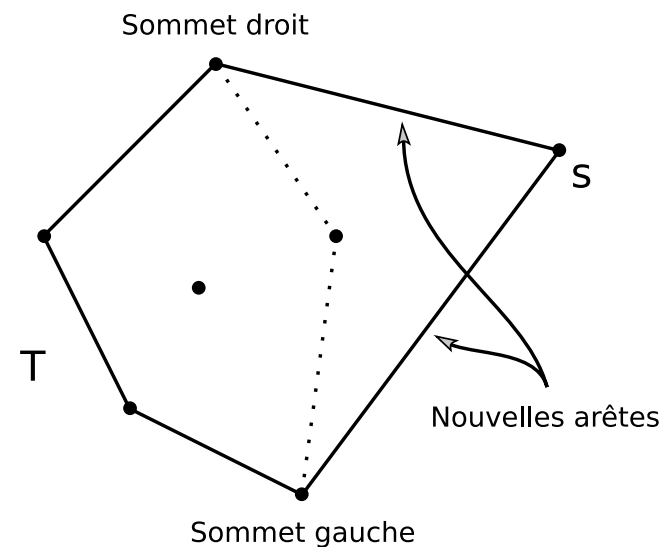
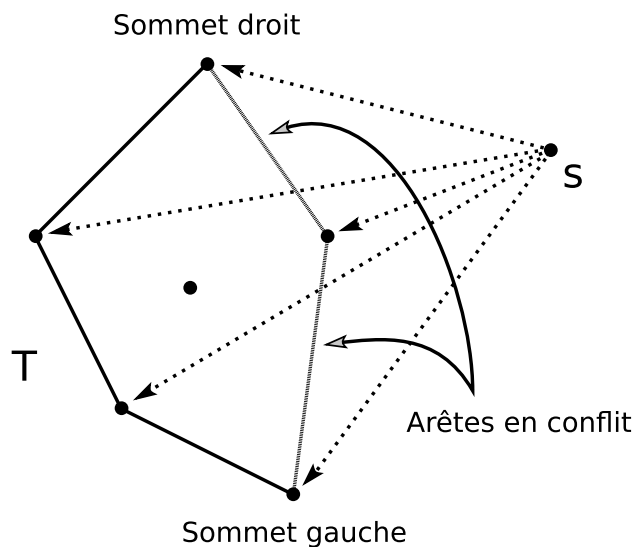
**Théorème** (du genre)

- (i)  $\chi$  est un entier pair.
- (ii)  $g$  est un entier naturel.

## 2.2. Algorithme incrémental de calcul de l'enveloppe convexe

---

- Au moins 3 points  $\rightarrow$  le premier triangle
- Insertion d'un nouveau point (représenté par un brin plongé) dans la carte représentant l'enveloppe convexe déjà construite



## 2.3. Opération d'insertion avec des cartes

---

- Description par induction structurelle sur la carte initiale
  - Indéterminisme dans le traitement des brins
  - Pas de stratégie de parcours
- Carte orientée avec des orbites non fermées
  - Moins de liaisons à casser
  - Orientation immédiate
- Repérage des arêtes en conflit

Elles sont contiguës et leurs *extrémités* sont uniques (propriété fondamentale mais non immédiate puisque l'algorithme traite séparément brins et liaisons) .

## 2.3. Opération d'insertion avec des cartes

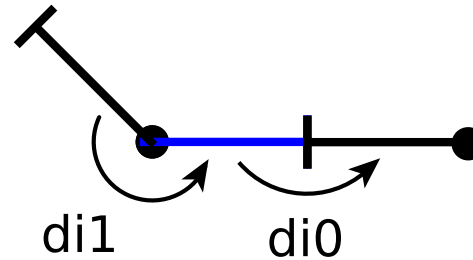
---

- Classification et propriétés de base des brins

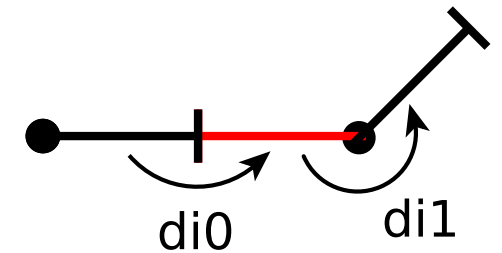
black\_dart



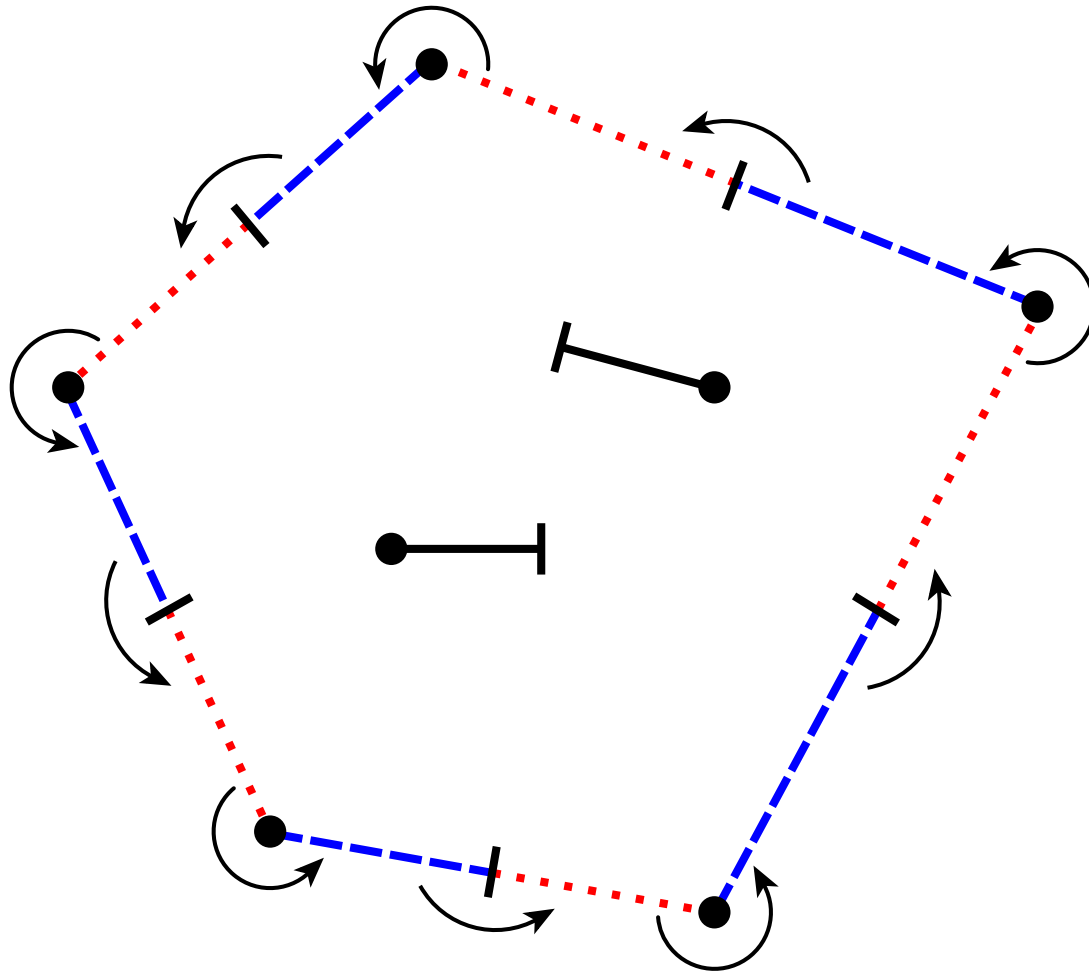
blue\_dart



red\_dart



- Initialement : noirs
- Nouveaux brins : rouges
- Au cours d'une étape d'insertion : passage de noir à bleu ou de bleu à noir



## 2.4. Structure de l'algorithme d'insertion en Coq

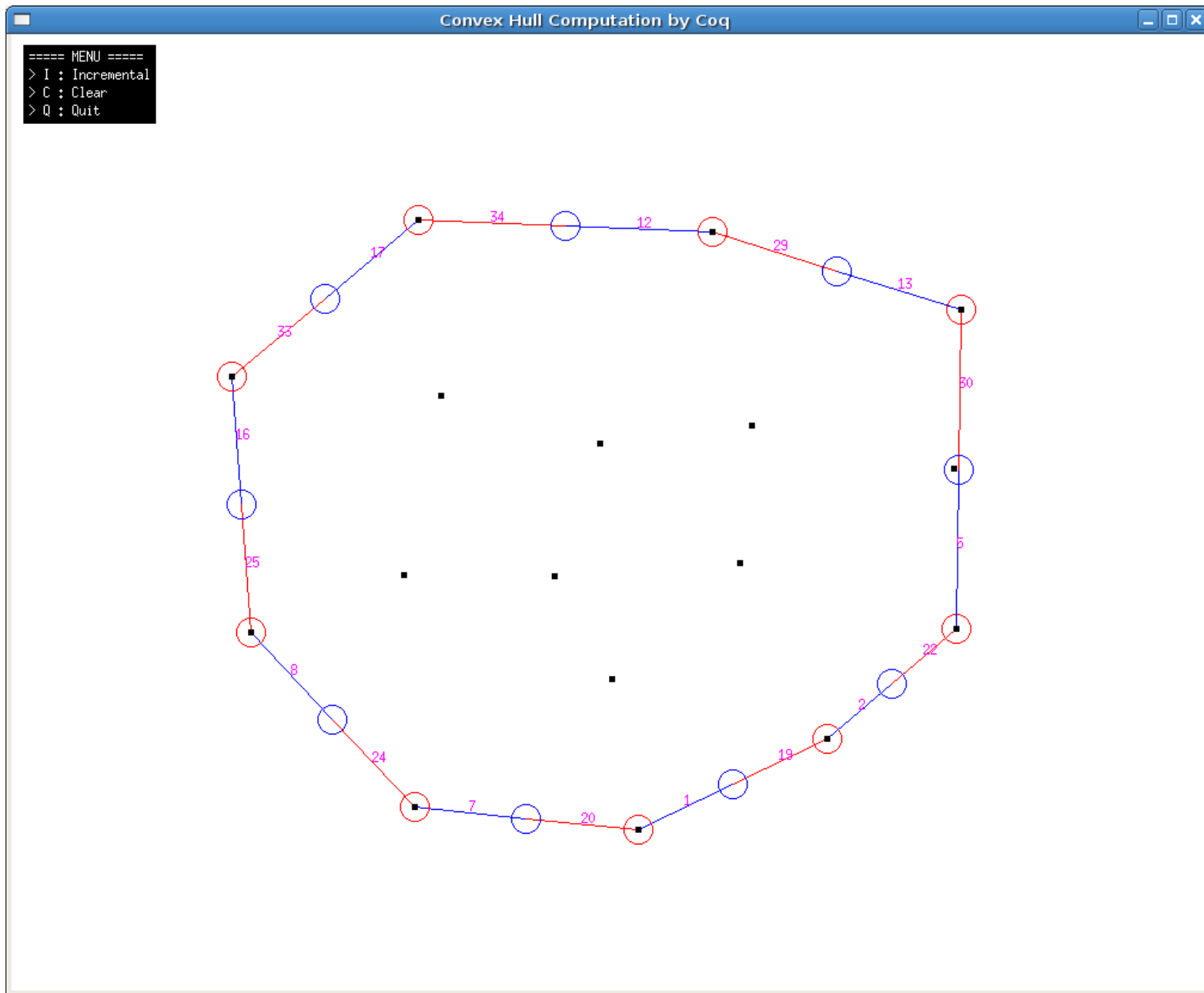
---

```
01: Fixpoint CHID (m:fmap) (mr:fmap) (x:dart) (p:point) (max:dart) struct m : fr
02: match m with
03:   V => I V x p
04: | I m0 x0 p0 =>
05:   if ( blue_dart_dec mr x0) then
06:     if (ccw_dec (fpoint mr x0) (fpoint mr (A mr di0 x0)) p) then
07:       (I (CHID m0 mr x p max) x0 p0)
08:     else if (left_dart_dec mr p x0) then
09:       (L (L (I (I (CHID m0 mr x p max) x0 p0) max p) di1 max x) di0
10:       else (I (CHID m0 mr x p max) x0 p0)
11:   else if ( red_dart_dec mr x0) then [...]
18: | L m0 di0 x0 y0 => [...]
22: | L m0 di1 x0 y0 => [...]
28: end.
```

### 3. Extraction et prototypage

---

- Extraction du langage de Coq vers le langage fonctionnel Ocaml  
On obtient une fonction des cartes vers les cartes `fmap` → `fmap`
- Passage d'une représentation réelle exacte à une représentation flottante
- Interface graphique et visualisation (v. transparent suivant)





## 4. Correction de l'algorithme

---

- Correction totale = terminaison + correction partielle
- Terminaison (imposée et donc garantie par Coq)
- Correction partielle
  - Aspects topologiques
  - Aspects géométriques

## 4.1. Propriétés à prouver

---

- Propriétés topologiques
  - conservation des brins initiaux (noirs et bleus)
  - conservation de l'invariant sur les cartes (préconditions)
  - après suppression des brins intérieurs (isolés) :
    - \* une seule composante connexe (si au moins 3 brins au départ)
    - \* 2 faces fermées (si au moins 3 brins au départ)
- Propriétés géométriques
  - Existence et unicité des extrémités gauche et droite pour les arêtes en conflit
  - être une enveloppe convexe

Tous les points sont à gauche de l'arête orientée  $xy$  de l'enveloppe obtenue.

## 4.2. Méthode de démonstration

---

- Preuves par induction structurelle sur les hypercartes
- Interdépendance entre géométrie et topologie
  - La bonne formation de la carte manipulée dépend de l'unicité des extrémités gauche et droite des arêtes en conflit.
  - Certaines branches de l'algorithme ne sont exécutées qu'une seule fois, ce qui garantit la bonne formation topologique de la carte manipulée (encore faut-il le prouver).
  - La mise en jeu du prédicat `ccw` fait intervenir presque immédiatement des considérations géométriques.

## 4.2. Méthode de démonstration

---

- Classification et évolution des couleurs des brins
  - Après chaque insertion, la couleur des brins est bien connue.
  - Pendant les appels récursifs, des liaisons ne sont pas reprises et la classification des brins doit être affinée.
  - Les couleurs dépendent notamment de la carte (carte en cours d'analyse structurelle ou carte de référence).
  - Une trentaine de lemmes pour suivre l'évolution de la couleur des brins au cours du calcul.

# Conclusions

---

- Modélisation géom. à base topologique et Spécifications formelles
- Description formelle et programmation d'un algorithme de calcul incrémental de l'enveloppe convexe,
  - basé sur des cartes combinatoires
  - programmé dans le langage fonctionnel du système Coq
- Terminaison (garantie par Coq) et premières preuves pour la correction partielle
- Extraction de Coq vers Ocaml  
et interface de saisie des données / visualisation du résultat
- Interdépendance forte entre topologie et géométrie

## Travaux en cours et perspectives

---

- Tirer parti de l'induction structurelle : l'algorithme fonctionne **efficacement** et sans stratégie de sélection des brins et des liaisons
- Tirer avantage des cartes : travailler à un **haut niveau d'abstraction** et utiliser l'induction noethérienne
- Généricité : difficulté du même travail sur un autre algorithme de calcul d'enveloppe convexe
- Vers des subdivisions plus complexes (Delaunay, Voronoï)
- Vers les dimensions supérieures