Topology of the Intersection of Two Parameterized Surfaces, Using Computations in 4D Space

Stéphane Chau, André Galligo

Topology of the Intersection of Two Parameterized Surfaces, Using Computations in 4D Space

Stéphane Chau and André Galligo
Université de Nice Sophia-Antipolis and INRIA (GALAAD project)
FRANCE

Résumé

The intersection curve of two parameterized surfaces is characterized by 3 equations $F_i(s,t) = G_i(u,v)$, $i = 1,2,3$ of 4 variables. So, it is the image of a curve in four dimensional space. We provide a method to draw such curve with a guaranteed topology.

1 Introduction

1.1 Interest of the problem

In Computer Aided Geometric Design (CAGD), parameterized surfaces are used for delimiting volumes. The computation of the intersection curve between such two surfaces is thus crucial for the description of the CAGD objects. A simple used method to address this problem consists in using a mesh for each surface, and then proceed to their intersection via intersection of triangles. A drawback is instability created by intersecting almost parallel triangles. A more stable method relies on global representations of the surfaces by B-splines ; however the usual CAGD procedures (offsetting, drafting, ...) do not conserve this model. In practice, so-called procedural surfaces (*i.e.* given by evaluation) are used, in CAGD systems, for representing sequences of constructions indicated by the user. Then a B-spline approximation is computed for further developments. So, even if the intersection method is exact, in its final step, it only provides an approximation of the "real" intersection curve.
Idealistically, approximations of the surfaces should not be separated from the intersection process. An intermediate strategy is to approximate the given surfaces by meshes of algebraic shapes more complex than the triangles ; hence the intersection locus will be more precise. A good choice is to approximate by Bézier surface patches of small degree (see section 1.2). Then, it is crucial to be able to efficiently intersect such two polynomial parameterized surfaces.
In this paper, we aim to contribute to a robust solution of this problem which

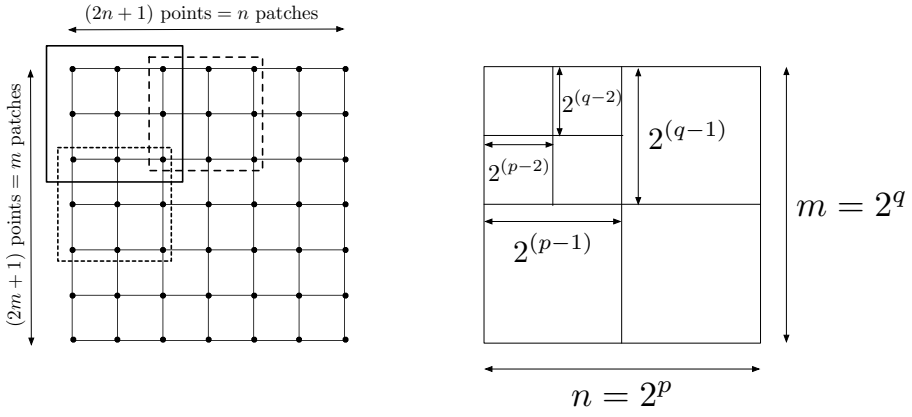

FIGURE 1 – Grid of biquadratic patches on the left. Grid of boxes with $n = 2^p$ and $m = 2^q$ on the right.

avoid some drawbacks as large intermediate algebraic expressions that appear in projection methods.
The intersection curve of two such parameterized surfaces is characterized by 3 equations $F_i(s,t) = G_i(u,v)$, $i = 1,2,3$ of 4 variables. So, it is the image of a curve in four dimensional space. We provide a method to draw such curve with a guaranteed topology.

1.2 An example of biquadratic meshing of a procedural surface

Let $\mathcal{S}$ be a general parameterized surface given by evaluations. We consider a grid of points on $\mathcal{S}$ of size $(2m+1, 2n+1)$. This is used to construct a grid of biquadratic patches of size (m,n). Figure 1 left illustrates this grid in the 2D parameter space. Thus the coefficients are shared between adjacent patches. An example of such kind of approximation is given in figure 2. In this example, we have on the left a shape composed by three B-spline surfaces, then we consider an offset, which cannot be represented by a B-spline, and we approximate it by a grid of 144 biquadratic patches (the result is shown on the right). In order to see the offset, a clipped picture is also given (figure 3).

Now, we consider two such grids and hierarchies on $\mathcal{S}_1$ and $\mathcal{S}_2$ two surfaces to be intersected. We produce another grid of $m \times n$ 3D boxes taking min-max values of the patch coefficients, each box contains the patch thanks to the convex hull property of the Bézier surfaces. Then, we build a quadtree hierarchy covering this grid. Figure 1 right illustrate this construction. Using these quadtrees we search for intersecting boxes and we obtain a set of pairs of intersecting boxes associated to patches. This process is efficient and, as we will

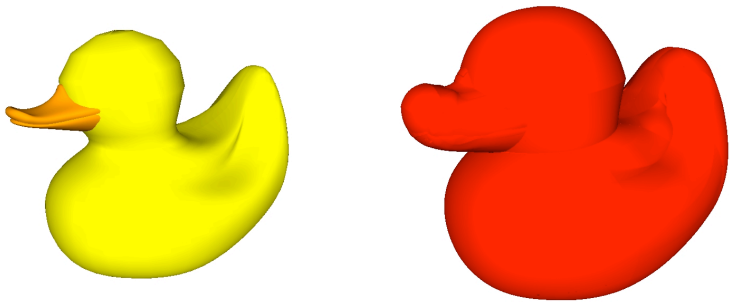

FIGURE 2 – Approximation of an offset by a grid of 144 biquadratic patches.

see, provides a good description of the intersection curve. However, it requires an efficient and robust algorithm for the intersection of two Bézier surface patches.

Remark 1 *If m and n as powers of 2, then the data structure is simplified.*

In the sequel of the paper, we concentrate on the presentation of our subdivision algorithm for the intersection of algebraic pacthes.

1.3 Organization of the paper

In section 2, a brief description of previous work on topology computation is given. Especially an introduction on subdivision approach for the plane curves is illustrated. Then, section 3 deals with the topology of an implicit four dimension curve. A complete description of its computation, by a subdivision method, is given in this case. In fact, this case corresponds to the intersection curve between two polynomial parameterized patches.
Some implementation aspects are addressed in section 4 and some examples are presented. The last section (5) is about the topology in $\mathbb{R}^3$. It shows that the link between the intersection problem in $\mathbb{R}^4$ and the corresponding geometric situation in $\mathbb{R}^3$ is not trivial.

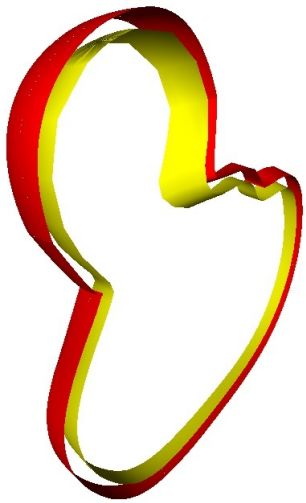

FIGURE 3 – Approximation of an offset by a grid of 144 biquadratic patches (clipped picture).

2 Previous work on topology computation of a curve

2.1 Isotopic curve

The topology of an algebraic curve $\mathcal{C}$ in $\mathbb{R}^n$ ($n \geq 2$) can be represented by a list of line segments whose concatenation forms a curve isotopic to $\mathcal{C}$. Several constructions make this definition effective.
Sweeping methods rely on parallel lines or planes and detect topological events (critical points) such as tangent points to the sweeping planes or singularities ; we refer to [1, 2] for planar curves and [3, 4] for spatial curves. With these algebraic approaches, the precise determination of the critical points generally requires to compute sub-resultant sequences and is often time consuming.

Subdivision and exclusion techniques (see [5, 6]) rely on (simple) criteria to remove unnecessary domains then restrict to domains where the situation is tame. Polynomial representation in Bernstein bases is generally preferred (see [7, 8]).

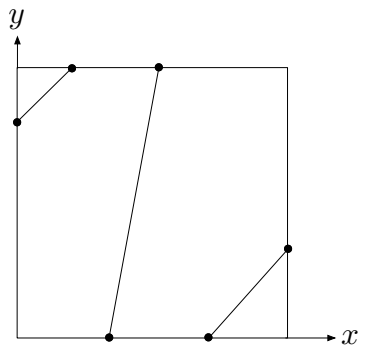

FIGURE 4 – Topology via regularity test in 2D case.

2.2 Regularity test and subdivision method

A subdivision approach for computing the topology of the intersection curve between two algebraic surfaces is given in [5]. It consists in subdividing the domain until a regularity test is satisfied. Let us briefly recall it.

Let $f(x, y)$ be a polynomial and $B = [a, b] \times [c, d] \subset \mathbb{R}^2$ a box, consider the implicit curve associated to f in the box B by the equation $f(x, y) = 0$. A regularity test will allow to determine uniquely the topology of the curve in the box from its intersection with the boundary. A collection of segments is provided, which realizes an isotopy.

Proposition 1 *If $\partial_y f(x, y) \neq 0$ for all $(x, y) \in B = [a, b] \times [c, d]$, then for all $x \in [a, b]$ there exists at most one $y \in [c, d]$ such that $f(x, y) = 0$.*

Proof. Let x_0 be a value in $[a, b]$. If there were two different values $y_0 < y_1$ in $[c, d]$ such that $f(x_0, y_0) = f(x_0, y_1) = 0$ then by Rolle's theorem, it would exist $y_2 \in [y_0, y_1]$ such that $\partial_y f(x_0, y_2) = 0$. □

Remark 2

- *This criterion considers $\partial_y f(x, y)$ for all values (x, y) in the box and not only for all points of the curve, so it is rather restrictive.*
- *To implement this criterion, the polynomial $\partial_y f(x, y)$ is expressed in Bernstein basis and the coefficients are required to share the same sign.*
- *A similar statement holds replacing the condition $\partial_y f(x, y) \neq 0$ by $\partial_x f(x, y) \neq 0$ (for all $(x, y) \in B$).*

If f satisfies this test, then the topology of the curve $\{(x, y) \in B \mid f(x, y) = 0\}$ can be determined uniquely knowing the intersection points between the curve and the border of B. Hence, a first step is to compute all these intersection

points (a point is repeated if its multiplicity is even) and sort them by their x component to obtain a list of points $p_1, p_2, \ldots, p_{2s-1}, p_{2s}$. Then, in the box, the curve is isotopic to the set of segments : $[p_1, p_2], \ldots, [p_{2s-1}, p_{2s}]$ (see the illustration in figure 4).
The criterion can be checked recursively subdividing the initial curve (using De Casteljau's algorithm) until a family of boxes is obtained where the test is verified.
The approach is extended (in [5]) to the case of 3D curve defined implicitly by 2 equations.
This provides an elegant and efficient solution to the topology computation problem of an intersection curve between two implicit surfaces.

3 Topology of a parameterized surface/parameterized surface intersection

3.1 Equations

Let F and G be two polynomial surface patches

$$F : \begin{pmatrix} [0,1]^2 & \longrightarrow & \mathbb{R}^3 \\ (s,t) & \longmapsto & F(s,t) \end{pmatrix}$$
$$G : \begin{pmatrix} [0,1]^2 & \longrightarrow & \mathbb{R}^3 \\ (u,v) & \longmapsto & G(u,v) \end{pmatrix}$$

We suppose that the intersection $F \cap G$ is a curve :

$$\mathcal{C} = \left\{ (s,t,u,v) \in [0,1]^4 \mid F(s,t) - G(u,v) = 0 \right\}.$$

Our aim is to compute the topology of $\mathcal{C}$ by a subdivision method generalizing the approach described in section 2. An injectivity criterion which says that for all $s_0 \in [0,1]$ there exist at most one $(t_0, u_0, v_0) \in [0,1]^3$ such that $F(s_0, t_0) - G(u_0, v_0) = 0$ is needed. So, for a fixed $s_0 \in [0,1]$, let us study the map :

$$\phi : \begin{pmatrix} [0,1]^3 & \longrightarrow & \mathbb{R}^3 \\ (t,u,v) & \longmapsto & F(s_0,t) - G(u,v) \end{pmatrix}$$

Thereafter, we set the notation $\phi(t,u,v) = F(s_0,t) - G(u,v) = (\phi_1, \phi_2, \phi_3)$.

3.2 Topology of a 4 dimension implicit curve (regularity criterion)

3.2.1 Construction of the injectivity criterion for ϕ

.

A necessary condition of injectivity is the local injectivity of ϕ. By the inverse function theorem, it is satisfied when the jacobian of ϕ is non zero over $[0,1]^3$:

$$\forall (t,u,v) \in [0,1]^3, \det\left(\partial_t \phi(t,u,v), \partial_u \phi(t,u,v), \partial_v \phi(t,u,v)\right) \neq 0.$$

If ϕ is not injective, there exist two different points A and B in $[0,1]^3$ such that for all $i \in \{1,2,3\}$, $\phi_i(A) = \phi_i(B)$. Our analysis relies on the introduction of the two following subsets of $[0,1]^3$:

$$S_1 := \left\{ M \in [0,1]^3 \mid \phi_1(M) = \phi_1(A) \right\} \tag{1}$$

$$C_{1,2} := \left\{ M \in [0,1]^3 \mid \phi_1(M) = \phi_1(A) \text{ and } \phi_2(M) = \phi_2(A) \right\}. \tag{2}$$

We assume local injectivity and look for sufficient conditions of injectivity of ϕ.

First case : A and B are on a same connected component of $C_{1,2}$ denoted by Γ. As Γ is a connected curve, local injectivity of ϕ implies that Γ can be parameterized (by the implicit function theorem). So ϕ_3 restricted to Γ is differentiable and takes the same value at A and B. Hence, ϕ_3 admits an extremum on $C_{1,2}$. This would contradicts local injectivity of ϕ, so it cannot happen.

Second case : A and B are on two different connected components of $C_{1,2}$ denoted by C_A and C_B. None of these two curves can describe a loop because this would contradict the local injectivity of ϕ.

Therefore C_A (respectively C_B) intersects two times the border of the cube $[0,1]^3$; in four distinct points P_1, P_2, P_3 and P_4. So we get a sufficient condition of injectivity if we can rule out this last possibility. Our strategy is to impose sufficient monotony conditions on ϕ_1 and ϕ_2.

3.2.2 Monotony condition on ϕ_1

First, we impose monotony conditions on ϕ_1 restricted to the edges of $[0,1]^3$. For example, we can require that ϕ_1 increases on each edges of $[0,1]^3$ as indicated in figure 5. So ϕ_1 vanishes at most once on each path going from the vertex O to the vertex I following the ordered egdes. This condition implies that the implicit surface S_1 (of equation $\phi_1(t,u,v) = \phi_1(A)$) is connected. Indeed, if S_1 admitted two connected components in the cube, they would intersect the edges at the same points which is impossible. Moreover we classify all possible configurations by the number of the intersection points (3, 4, 5 or 6) between S_1 and the edges as illustrated in figure 6. Note that as $C_{1,2} \subset S_1$ and $\partial S_1 \subset \partial[0,1]^3$, the equality $\#(C_{1,2} \cap \partial S_1) = \#(C_{1,2} \cap \partial[0,1]^3)$ holds (see figure 7)

3.2.3 Monotony condition on ϕ_2 along ∂S_1

Now, we study each configuration. We impose monotony conditions on ϕ_2 along the border of S_1 to force $C_{1,2}$ to have at most two intersection points with this border.
The following lemma will be useful :

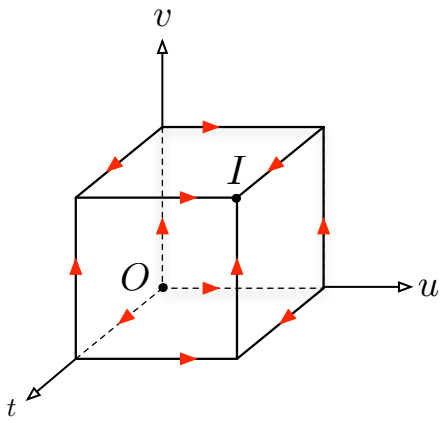

FIGURE 5 – Example of monotony of ϕ_1 on the edges of $[0,1]^3$.

Lemma 1 *Let f be a $\mathcal{C}^1$ real function over an open convex set $\mathcal{U} \subset \mathbb{R}^2$ and h be a nonzero vector in $\mathbb{R}^2$. If for all $u \in \mathcal{U}$ we have $\nabla f(u) \cdot h > 0$, then f is increasing in the direction h on $\mathcal{U}$ i.e $\forall u \in \mathcal{U}$ and $\forall \epsilon > 0$ such that $(u + \epsilon\, h) \in \mathcal{U}$, we have $f(u + \epsilon\, h) > f(u)$.*

Proof. Let $u_0 \in \mathcal{U}$ and $\epsilon > 0$ such that $(u + \epsilon\, h) \in \mathcal{U}$. Then $f(u_0 + \epsilon\, h) - f(u_0) = \int_0^1 \varphi(t)\, dt$ with $\varphi(t) = \nabla f\,(u_0 + t\epsilon h) \cdot h$ which is positive. □

— Replacing f by $-f$, we get similarly that $\nabla f(u) \cdot h < 0$ implies f is decreasing in the direction h on $\mathcal{U}$.

— Recall that in the plane, for a nonzero vector $\overrightarrow{w} = (a, b)$, the vector $\overrightarrow{w}^{\perp} := (-b, a)$ is normal to $\overrightarrow{w}$ and the oriented angle $(\overrightarrow{w}, \overrightarrow{w}^{\perp})$ is equal to $\pi/2$.

So, in order to ensure monotony of ϕ_2 along the border of S_1 (see figure 7), we orient S_1 by the vector field $\nabla\phi_1$. This induces an orientation on the border ∂S_1 of S_1 ; ∂S_1 is the intersection of S_1 with the faces of the cube. This orientation of the border of S_1 in each face is given by $\overrightarrow{w}^{\perp}$ where $\overrightarrow{w}$ is the projection of $\nabla\phi_1$ on the faces. Then, we impose a monotony direction of ϕ_2 restricted to ∂S_1 on each face of the cube. To illustrate this procedure, figure 8 represents, in the three coordinates planes, the monotonies shown on figure 7 : the desired monotony in the (u, v)-plane (pictured in the middle of figure 8) is obtained by projecting the vector $\nabla\phi_1$ on this plane and we have $\overrightarrow{w} = (\partial_u\phi_1(0, \cdot), \partial_v\phi_1(0, \cdot))$. Then, we force the decreasing of $(u, v) \mapsto \phi_2(0, u, v)$ in the direction $\overrightarrow{w}^{\perp}$. Applying Lemma 1, we require :

$$\forall (u, v) \in [0,1]^2, \begin{pmatrix} \partial_u\phi_2(0,u,v) \\ \partial_v\phi_2(0,u,v) \end{pmatrix} \cdot \begin{pmatrix} -\partial_v\phi_1(0,u,v) \\ \partial_u\phi_1(0,u,v) \end{pmatrix} < 0.$$

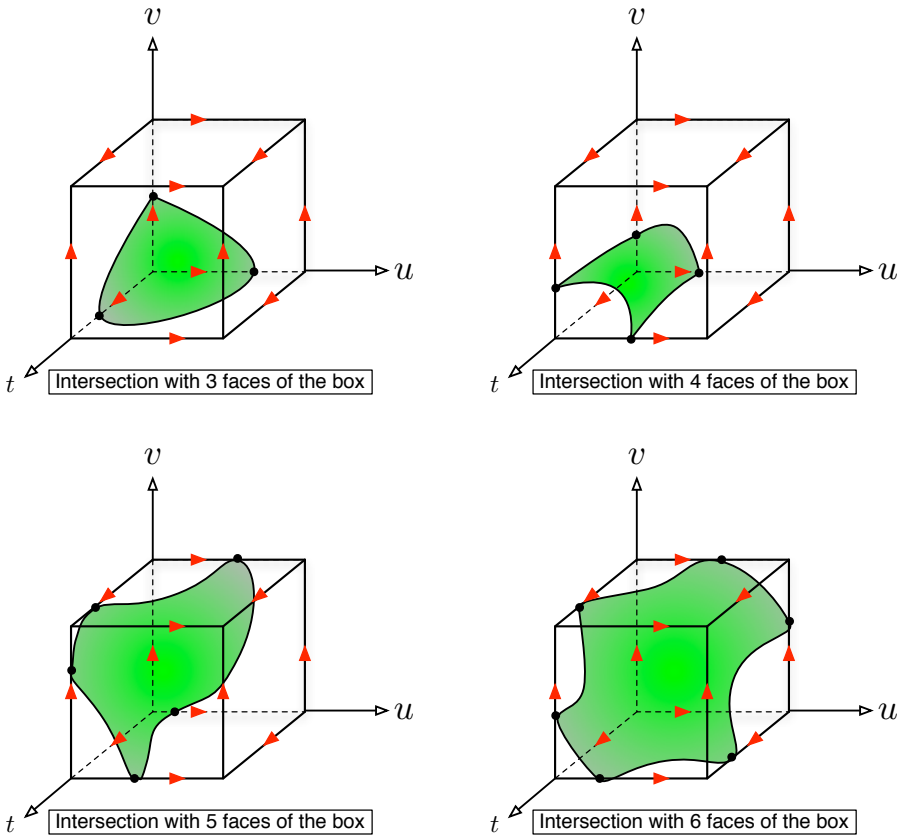

FIGURE 6 – Configurations of the surface S_1 in $[0,1]^3$ under the monotony constraint on ϕ_1.

This previous dot product is a polynomial of bi-degree $(3,3)$ with respect to the variables (u,v). Considered also as a polynomial in s (that we fixed at the beginning of this section) it is of degree 4 in s.

3.2.4 Choice of monotony constraints

Here, we present our choice of sufficient condition such that $\#(C_{1,2} \cap \partial S_1) \leq 2$. First, we consider the case where S_1 intersects the 6 faces of the cube $[0,1]^3$. In the other cases, we just skip the condition corresponding to missing segments contracted to a point (see figure 9).

The border of S_1 is isotopic to an hexagon $\{M_1, \ldots, M_6\}$ as shown on figure

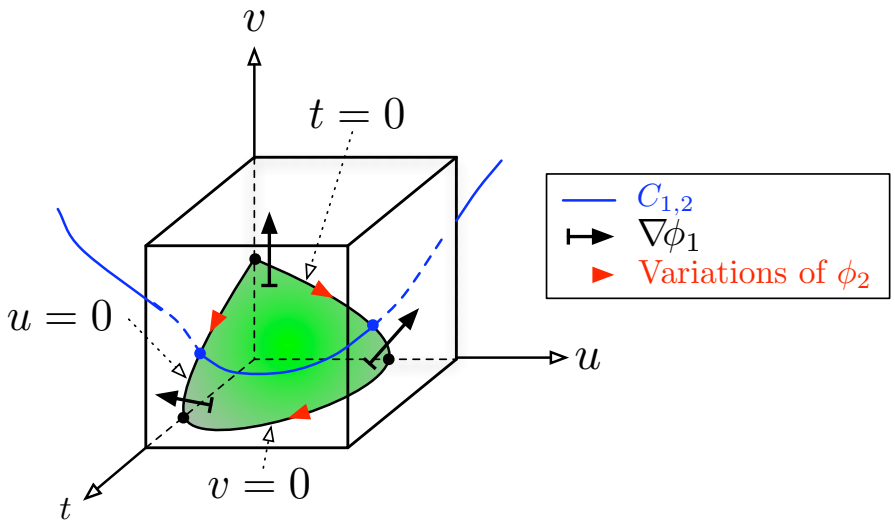

FIGURE 7 – Example of monotony of ϕ_2 along the border of S_1.

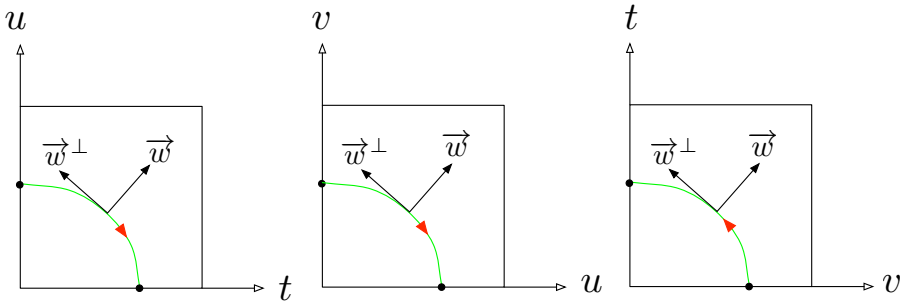

FIGURE 8 – Traces of S_1 on the faces of the box with orientations.

11.
A sufficient monotony condition is given by a choice of an initial point M_I and a final point M_F among $\{M_1, \ldots, M_6\}$ with the possible choice $M_I = M_F$ such that ϕ_2 is monotonic on the paths on ∂S_1 joining M_I to M_F. This clearly implies that ϕ_2 vanishes at most twice on ∂S_1. Now, we can extend our choice of sufficient conditions simply by remarking that the 4 variables $\{s, t, u, v\}$ play similar roles.

1. Instead of fixing s, we can fix t, u or v and consider the corresponding maps.
2. Also, the roles of ϕ_1, ϕ_2,and ϕ_3 can be exchanged.

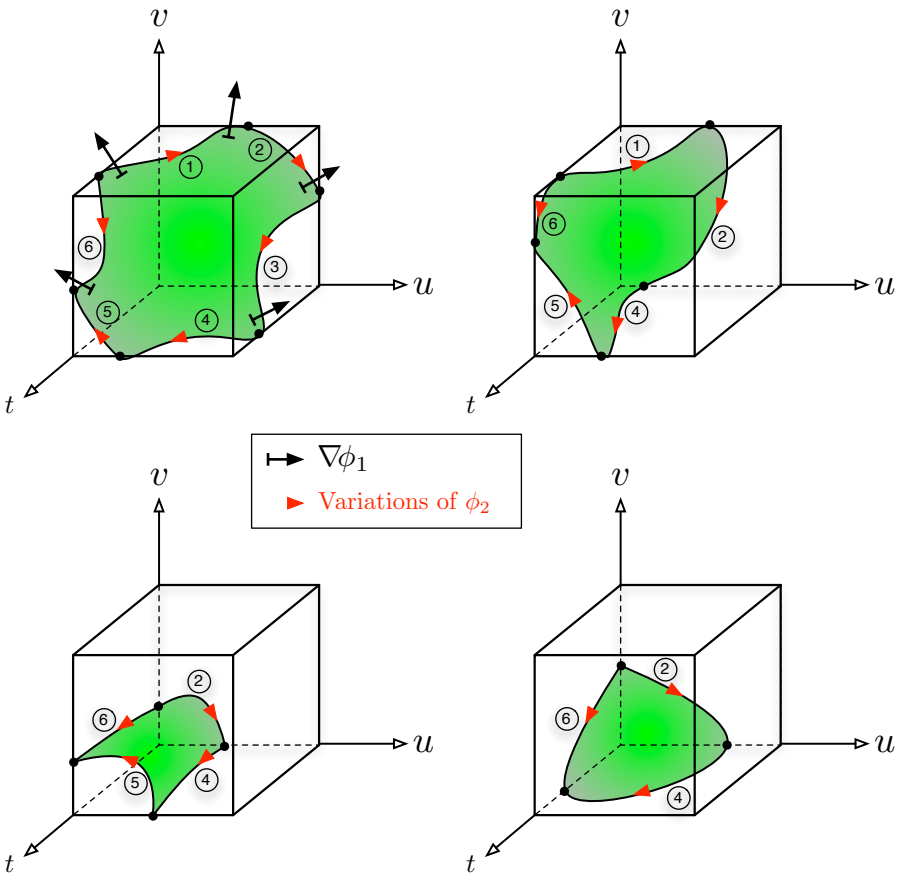

FIGURE 9 – Exemple of monotony of ϕ_2 along the border of S_1 in the case where S_1 intersect 6 faces of the box and the resulting configurations in the other cases.

All these options will be considered to speed up the implementation.

4 Algorithms and data structure used for implementation

In this section, we present some implementation aspects of the intersection algorithm described in section 3. They are implemented in Axel [1] which is an algebraic geometric modeler.

1. http ://axel.inria.fr

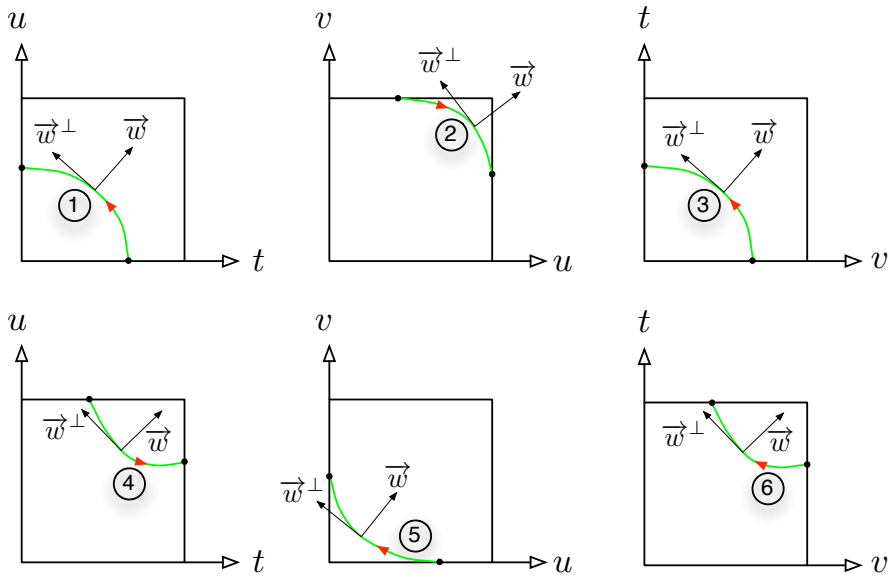

FIGURE 10 – Traces of S_1 on the faces of the box (it corresponds to the case represented in figure 9).

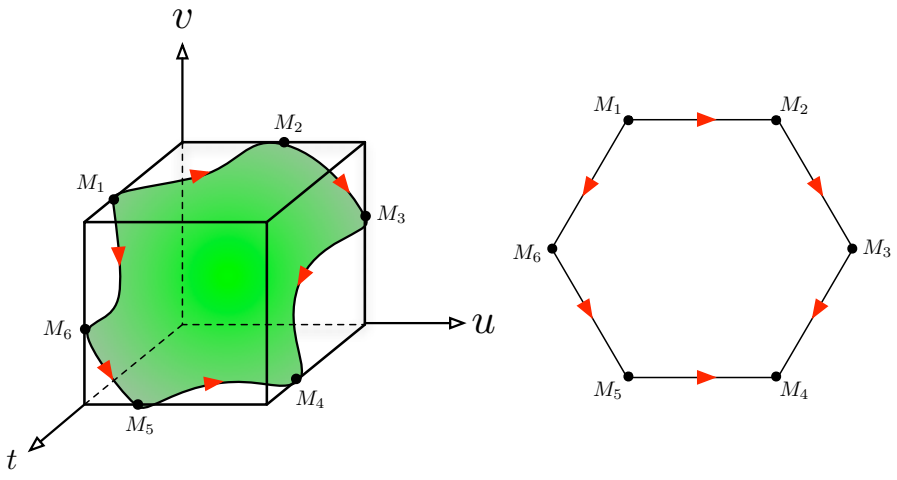

FIGURE 11 – Traces of S_1 on the faces of the box (it corresponds to the case represented in figure 9).

4.1 Hexatree data structure and topology

A subdivision algorithm on a box in $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3] \times [a_4, b_4] \subset \mathbb{R}^4$ explores sub-boxes constructed by considering intermediate values c_i between

a_i and b_i for $i \in \{1, \ldots, 4\}$; here we choose $c_i = \dfrac{a_i + b_i}{2}$. So a box has 16 sub-boxes. Iterating this construction, an hexatree is build ; *i.e.* each node of the tree has 16 children numbered from 0 to 15. In binary expression, this number is written $\alpha_1\alpha_2\alpha_3\alpha_4$ with $\alpha_i = 0$ or 1 ; for $i \in \{1, \ldots, 4\}$, if $\alpha_i = 0$, the sub-boxe is constructed over $[a_i, c_i]$ and if $\alpha_i = 1$ it is constructed over $[c_i, b_i]$. For example, the child twelve is written 1100 and corresponds to the sub-box $[c_1, b_1] \times [c_2, b_2] \times [a_3, c_3] \times [a_4, c_4]$.
This is called an hexatree data structure, it generalizes the quadtrees which are widely used to represent planar shapes. To each node of the tree is associated a label which stores the needed information. Here, the information will be the description of the topology of the intersection curve $\mathcal{C}$ into the corresponding sub-boxe. More precisely, we require that, at the leaves of the tree, this intersection is empty or its dimensions are below some threshold or it is isotopic to a collection of disjoint segments ; each segment connects two intersection points of the curve $\mathcal{C}$ with the border of the considered sub-box. Each such segment is represented by the coordinates of its extremal points. Note that in $\mathbb{R}^4$, all the 16 children sub-boxes of a given box are adjacent. Our injectivity criterion described in 3 is implemented in a test function (called ***regular***) if it returns false on a sub-box then the sub-box is subdivided.

4.2 Subdivision algorithm

The algorithm 4.1 describes the subdivision method for the topology computation. Some other functions are needed and are described in the sequel.

Algorithm 4.1: Subdivision algorithm for topology in 4D.

`topology`$(\mathcal{C}, B, \epsilon)$
Input: The curve $\mathcal{C}$, a box $B = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3] \times [a_4, b_4]$ and a tolerance ϵ.
Output: A liste of segments in $\mathbb{R}^4$ representing the topology.
Create the hexatree $\mathcal{H}$;
Initialize the root of $\mathcal{H}$ by B and the intersection points $\mathcal{C} \cap \partial B$;
Create a list of nodes $\mathcal{L}$;
$\mathcal{L} \longleftarrow$ `rootOf`$(\mathcal{H})$;
while $\mathcal{L} \neq \emptyset$ **do**
  Take the first item n of $\mathcal{L}$ (and remove it from $\mathcal{L}$);
  if `regular`$(\mathcal{C}, n)$ **then**
    $n \longleftarrow$ `regularTopology`$(\mathcal{C}, n)$;
  else if *the current box has a size* $\geq \epsilon$ **then**
    $\mathcal{L} \longleftarrow$ `subdivision`$(\mathcal{C}, n)$;
  else
    Give an arbitrary topology by connecting all the border points to the center of the box (this applies when we stop the subdivision);
  end
end
return `fusion`$(\mathcal{H})$;

Now we describe the other functions called by `topology`.

Function `regular` :

This function is the injectivity criterion described in section 3. In fact there are 4 different tests and each of them corresponds to the fixed variable choice s, t, u or v (see algorithm 4.2). If one is verified, then we call the corresponding function `regularTopology`.

Function `regularTopology` :

If one of the four regularity tests `regular` is verified, then the topology of $\mathcal{C}$ is known. In the function `regularTopology`, we just have to connect the border points in the current node. In fact, we also have 4 different `regularTopology` functions corresponding to the fixed variable s, t, u or v. For example, if $s = s_0$ is fixed, then we have, in the current node, a list of even number of border points $p_1, p_2, \ldots, p_{2k-1}, p_{2k}$ (by repeating a point if its multiplicity is even) sorted by their s component. Then, the topology is described by the list of segments : $[p_1, p_2], \ldots, [p_{2k-1}, p_{2k}]$.

Function `subdivision` :

This function subdivides the current box creating 16 children as described in section 4.1. It allocates the inherited intersection points and compute the new

Algorithm 4.2: Injectivity criterion.

```
regular(C, n)
if φ is locally injective in n then
    if φ_1 has the wanted monotony on the edges of n then
        if φ_2 or φ_3 has the wanted monotony on ∂S_1 then
            return true;
        else
            return false;
        end
    else if φ_2 has the wanted monotony on the edges of n then
        if φ_1 or φ_3 has the wanted monotony on ∂S_2 then
            return true;
        else
            return false;
        end
    else if φ_3 has the wanted monotony on the edges of n then
        if φ_1 or φ_2 has the wanted monotony on ∂S_3 then
            return true;
        else
            return false;
        end
    else
        return false;
    end
else
    return false;
end
```

intersection points that appear with the faces of these sub-boxes.

Function `fusion` :

This function is called when the construction of the hexatree $\mathcal{H}$ is finished. More precisely each leaf of $\mathcal{H}$ contains the topology in the corresponding sub-box. `fusion` provides the topology of $\mathcal{C}$ in the initial box B. Its implementation (see algorithm 4.3) consists in merging recursively the topology between the children of each node. For a given node n and an integer $i \in \{0, \dots, 15\}$, we denote by `child`(i, n) the i-th child as described in section 4.1. Besides, if l_1 and l_2 are two list of segments in $\mathbb{R}^4$, `merge`(l_1, l_2) will be the list of segments in $\mathbb{R}^4$ formed by all the segments of l_1 and l_2.

Algorithm 4.3: Topology by subdivision.

```
fusion(n)
Input: A node of hexatree as it is described in section 4.1
Output: A list of segments in ℝ⁴
if n is a leaf then
  return the list of segments in n;
else
  return
  merge(
    merge(
      merge(
        merge(
          merge(fusion(child(0, n)), fusion(child(1, n)))
          merge(fusion(child(2, n)), fusion(child(3, n))))
        merge(
          merge(fusion(child(4, n)), fusion(child(5, n)))
          merge(fusion(child(6, n)), fusion(child(7, n)))))
      merge(
        merge(
          merge(fusion(child(8, n)), fusion(child(9, n)))
          merge(fusion(child(10, n)), fusion(child(11, n))))
        merge(
          merge(fusion(child(12, n)), fusion(child(13, n)))
          merge(fusion(child(14, n)), fusion(child(15, n))))));
end
```

4.3 Connected components and loops

The algorithm 4.1 allows to identify the connected components easily. Indeed, the resulted topology of $\mathcal{C}$ is a list of segments (in $\mathbb{R}^4$) of the form $\{[p_1, p_2], \ldots, [p_{2k-1}, p_{2k}]\}$ where k is a positiv integer. If there exist $i \in \{1, \ldots, k-1\}$, such that $p_{2i} \neq p_{2i+1}$, then the two segments $[p_{2i-1}, p_{2i}]$ and $[p_{2i+1}, p_{2i+2}]$ are on two different connected components of the topology. A similar simple argument allows to detect the loops (connected) components.

4.4 Examples

We illustrate the algorithm on some examples. First we gives two intersection situations of two polynomial patches shown on figures 12 and 13. Another classical example (the teapot) is given. Figure 14 shows an approximation of the teapot by 32 biquadratic patches with intersection loci. The resulting topology of these loci is shown on figure 15.

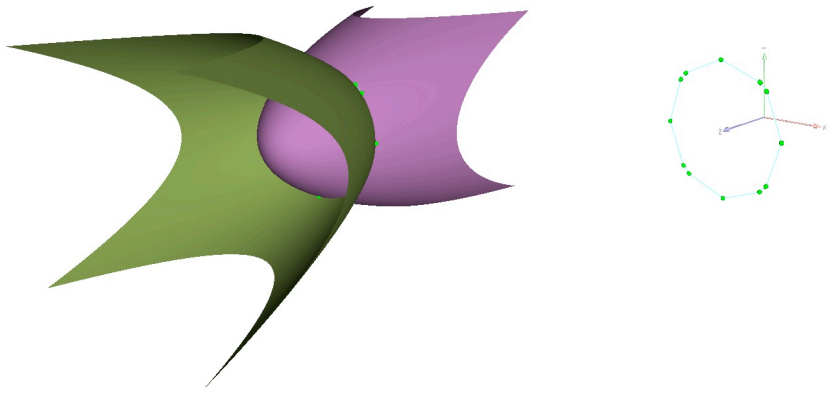

FIGURE 12 – Example of intersection between two polynomial patches.

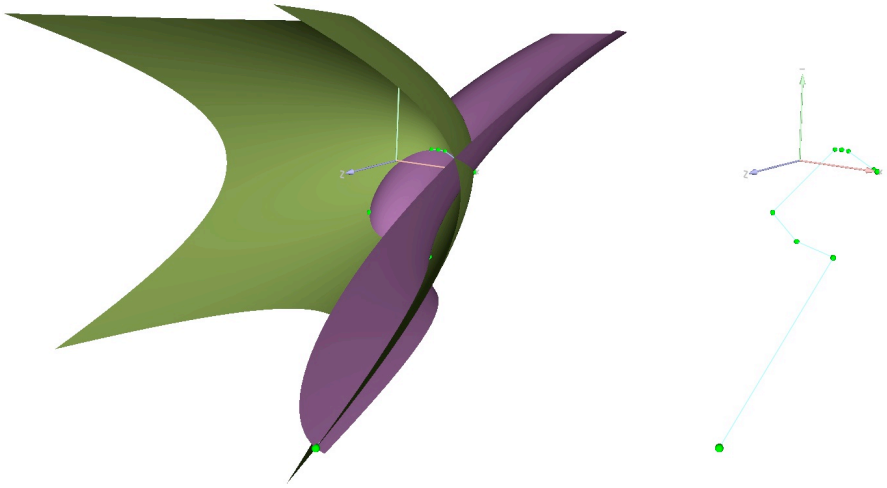

FIGURE 13 – Example of intersection between two polynomial patches.

5 Topology in $\mathbb{R}^3$

Sections 3 and 4 presented an algorithm for computing the topology of a curve $\mathcal{C}$ in $\mathbb{R}^4$ defined by 3 equations $F(s,t) = G(u,v)$ (with $(s,t,u,v) \in [0,1]^4$

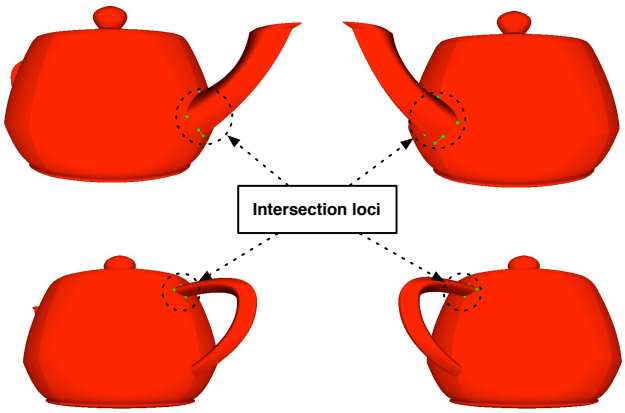

FIGURE 14 – Teapot intersection loci.

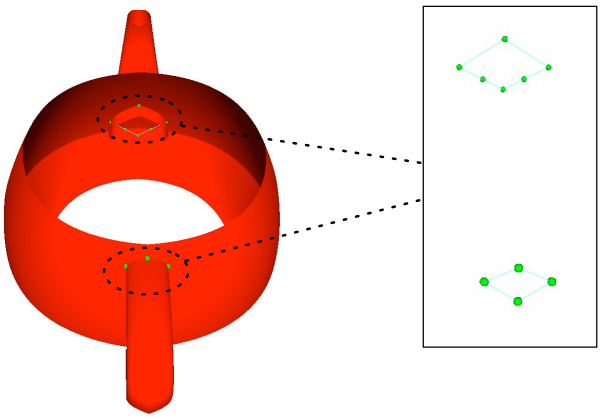

FIGURE 15 – Topology of the teapot intersection.

for example). We introduce the following notations :

$$\pi_1 : \begin{pmatrix} [0,1]^4 & \longrightarrow & \mathbb{R}^2 \\ (s,t,u,v) & \longmapsto & (s,t) \end{pmatrix}$$
$$\pi_2 : \begin{pmatrix} [0,1]^4 & \longrightarrow & \mathbb{R}^2 \\ (s,t,u,v) & \longmapsto & (u,v) \end{pmatrix}.$$

The intersection Γ in $\mathbb{R}^3$ of the two parameterized surface patches F and G is the image of $\mathcal{C}$ by $F \circ \pi_1$ (or $G \circ \pi_2$) of $\mathcal{C}$. Our algorithm guarantee (up to the tolerance ϵ) the topology of $\mathcal{C}$, which is isotopic to a collection of segments in $[0,1]^4$. This implies that the image by $F \circ \pi_1$ of a connected component $\mathcal{C}_1$ of $\mathcal{C}$ is connected. However, if $\mathcal{C}_1$ is a loop in $]0,1[^4$ (a closed path) then its image is also a loop in $\mathbb{R}^3$ but we do not know its knot structure. Moreover, if $\mathcal{C}$ admits several connected components which are loops in $[0,1]^4$, their images by $F \circ \pi_1$ in $\mathbb{R}^3$ may be interlaced (like the olympic rings). If $\mathcal{C}_1$ is determined by a segment discretization which is too coarse, the knot structure (and the interlacements) can be missed in the image by $F \circ \pi_1$ of this piecewise approximation. We may have the situation depicted in figure 16.

Similarly, the topology of the projection $\widehat{\mathcal{C}}$ of $\mathcal{C} \subset [0,1]^4$ on $[0,1]^2$ by π_1, may not be determined by a coarse discretization of $\mathcal{C}$, even if this discretization is sufficient to determine the topology of $\mathcal{C}$ in $[0,1]^4$, see figure 17 : the self-intersection point is missed. In order to capture these features, the algorithm described in section 3 and 4 should be extended and the subdivision criteria refined.

As described in sections 3 and 4, we chose a threshold ϵ such that the singular points of the curve Γ will be contained in boxes of size smaller than ϵ. We aim to determine the topology of the curve Γ up to this indetermination, *i.e.* two

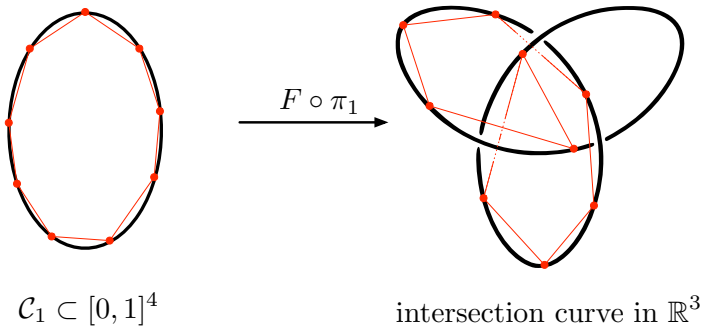

FIGURE 16 – Image of a loop with knot structure.

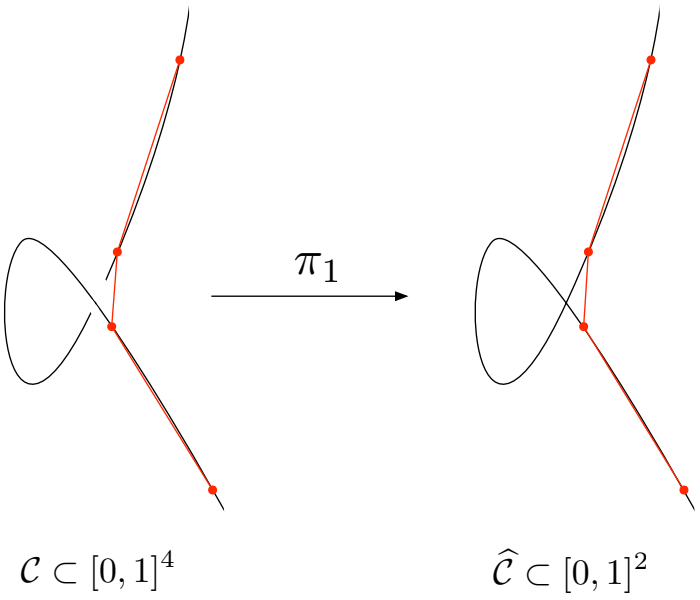

FIGURE 17 – Missing a self-intersection point by projection.

segments entering a box of size smaller than ϵ are supposed to intersect and form a singular point. All other points are considered smooth.
Suppose that $\mathcal{C}$ has k loops connected components (where k is a positiv integer) denoted respectively by $\mathcal{C}_1, \ldots, \mathcal{C}_k$ (we can detect them by using the criterion described in section 4.3). We denote $\Gamma_i = (F \circ \pi_1)(\mathcal{C}_i)$ for all $i \in \{1, \ldots, k\}$.

5.1 One curve box

Recall that each node of the hexatree $\mathcal{H}$ (described in section 4.1) stores a box in $\mathbb{R}^4$ and the topology of $\mathcal{C}$ in this box. Let n be a node of $\mathcal{H}$, $\mathcal{B}_n$ be the corresponding box and $B_n = B_F \cap B_G$, where B_F (respectively B_G) is the bounding box constructed with the control points of $F(s,t)$ (respectively $G(u,v)$) written in the Bernstein basis with respect to $\pi_1(\mathcal{B}_n)$ (respectively $\pi_2(\mathcal{B}_n)$). Then, by the convex hull property of the Bézier patches, the bounding box B_n contains the part of Γ corresponding to $\mathcal{B}_n$ *i.e.* the image of $\mathcal{C} \cap \mathcal{B}_n$ by $F \circ \pi_1$ (or $G \circ \pi_2$).

The discretization of $\mathcal{C}$ is refined, by subdividing all the leafs of $\mathcal{H}$, such that each box (in $\mathbb{R}^4$) intersecting one of the loops $\mathcal{C}_1, \ldots, \mathcal{C}_k$ contains at most one segment, *i.e.* its border intersects $\mathcal{C}$ in two points. Note that in the previous

section our algorithm allowed more intersection points. After this step some ambiguities of the node and interlacement structure of Γ may remain. One can see in figure 18 two bounding boxes (in $\mathbb{R}^3$) sharing interior points. Joining the pairs of points on the borders, the red curve segment may (or may not) pass behind the other green curve segment. So we need to refine further the discretization.

Lemma 2 *Let $\gamma_1, \gamma_2 \subset \Gamma$ be two disjoint segments of curves. After a finite number of subdivisions of $(F \circ \pi_1)^{-1}(\gamma_1)$ and $(F \circ \pi_1)^{-1}(\gamma_2)$, the boxes containing γ_1 are disjoint from the boxes containing γ_2.*

Proof. Indeed by subdivision, the boxes can be made nearer to the curves than the distance between the two curves. □

The subdivision on the leafs of $\mathcal{H}$ is refined by using lemma 2. Then, we rule out potential ambiguity on interlacements between two loops (situation corresponding the to right picture on figure 16) because we avoid the situation depicted on figure 18. So it remains to analyze the ambiguity on a possible node that is not a loop.

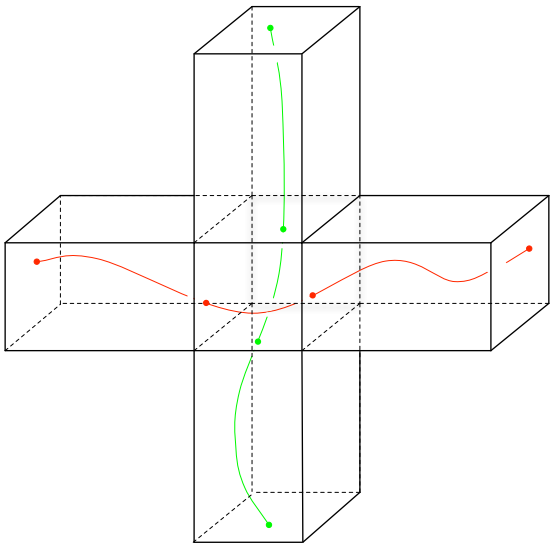

FIGURE 18 – Two boxes sharing interior points.

5.2 Node and discretization

Lemma 3 *Let $\gamma \subset \Gamma$ be a segment of curve contained in a bounding box obtained after the subdivision process described in section 5.1. Then, the border of this box has just two points p_1 and p_2 of γ. After a finite number of subdivisions of $(F \circ \pi_1)^{-1}(\gamma)$, we have $\det(N_F, N_G, \overrightarrow{p_1p_2}) \neq 0$ (in the corresponding box) with $N_F = \partial_s F \times \partial_t F$ and $N_G = \partial_u G \times \partial_v G$.*

Proof. The condition $\det(N_F, N_G, \overrightarrow{p_1p_2}) \neq 0$ means that the tangent vector of γ is never orthogonal to $\overrightarrow{p_1p_2}$. As γ is smooth par hypothesis, the lemma is a consequence of the implicit function theorem. □

If we subdivide the leafs of $\mathcal{H}$ by using lemma 3, then we rule out potential interlacements ambiguities inside each bounding box. However, it remains to avoid interlacing from two adjacent branches.

Proposition 2 *Assume the discretization satisfies lemma 2 and lemma 3. Suppose also that $\det(N_F, N_G, \overrightarrow{p_1p_2}) \neq 0$, $\det(N_F, N_G, \overrightarrow{p_2p_3}) \neq 0$ and $\det(N_F, N_G, \overrightarrow{p_1p_3}) \neq 0$ for two adjacent branches $[p_1, p_2]$ and $[p_2, p_3]$.*

If the image (by $F \circ \pi_1$ or $G \circ \pi_2$) of a loop connected component of $\mathcal{C}$ admits a node, then it shows up on the discretization i.e. *the sequence of segments obtained by subdivision also describes a node isotopic to that of Γ.*

Proof. Indeed, we will have the situation depicted on figure 19. By the conditions $\det(N_F, N_G, \overrightarrow{p_1p_2}) \neq 0$, $\det(N_F, N_G, \overrightarrow{p_2p_3}) \neq 0$ and $\det(N_F, N_G, \overrightarrow{p_1p_3}) \neq 0$, we cannot have a node with formed by two adjacent segments (depicted on figure 20). So, we just have to investigate the case where we have at least three segments $[p_1, p_2]$, $[p_2, p_3]$ and $[p_3, p_4]$. Lemma 3 ensure that each of these segments does not interlace. If the three segments are interlacing, then the bounding boxes containing respectively $[p_1, p_2]$ and $[p_3, p_4]$ intersect each other so it contradicts lemma 2. □

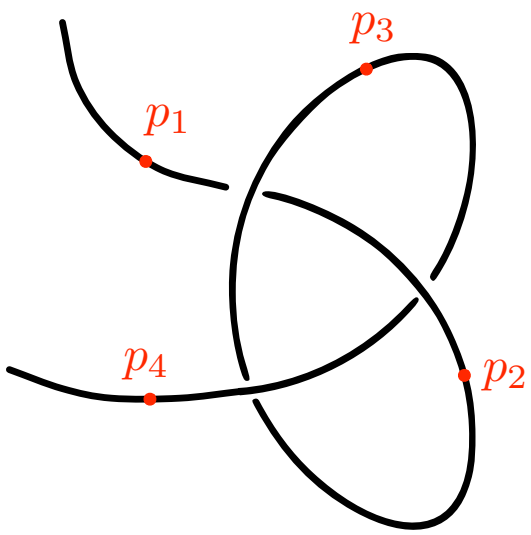

FIGURE 19 – Interlacement situation.

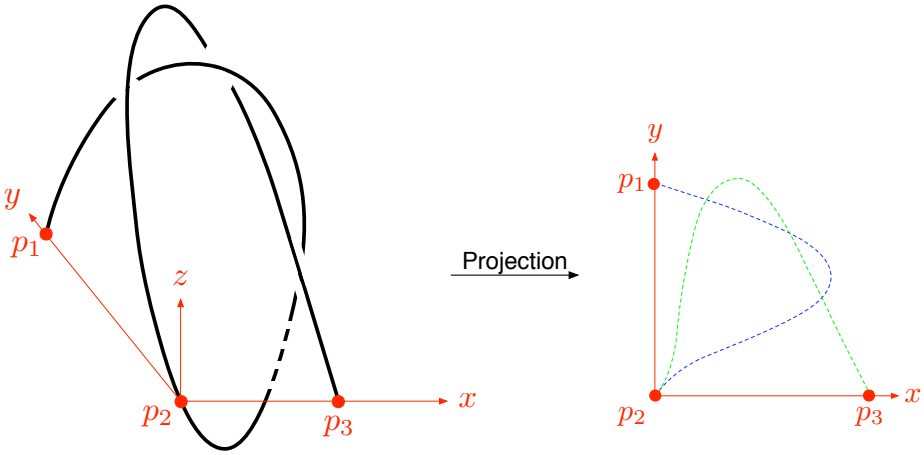

FIGURE 20 – Interlacement with two adjacent segments.

Références

[1] L. González-Vega, I. Necula, Efficient topology determination of implicitly defined algebraic plane curves, Comput. Aided Geom. Design 19 (9) (2002) 719–743.

[2] T. A. Grandine, F. W. Klein, A new approach to the surface intersection problem, Computer Aided Geometric Design 14 (1997) 111–134.

[3] J. G. Alcázar, J. R. Sendra, Computing the topology of real algebraic space curves, J. Symbolic Comput. 39 (2005) 719–744.

[4] G. Gatellier, A. Labrouzy, B. Mourrain, J.-P. Técourt, Computing the topology of 3-dimensional algebraic curves, in : Computational Methods for Algebraic Spline Surfaces, Springer-Verlag, 2004, pp. 27–44.

[5] C. Liang, B. Mourrain, J.-P. Pavone, Subdivision methods for the topology of 2d and 3d implicit curves, in : B. J. . R. Piene (Ed.), Computational Methods for Algebraic Spline Surfaces, 2005, Computational Methods for Algebraic Spline Surfaces, Springer, Oslo, Norway, 2007, pp. 171–186.
URL `http://hal.inria.fr/inria-00130216/en/`

[6] S. Plantinga, G. Vegter, Isotopic approximation of implicit curves and surfaces, in : SGP '04 : Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, ACM Press, New York, NY, USA, 2004, pp. 245–254.

[7] G. Farin, Curves and Surfaces for Computer Aided Geometric Design : A Practical Guide, 3rd Ed., Academic Press, 1993.

[8] T. W. Sederberg, Algorithm for algebraic curve intersection, Comput. Aided Des. 21 (9) (1989) 547–554.

[9] S. Chau, M. Oberneder, A. Galligo, B. Juttler, Intersecting biquadratic bézier surface patches, in : B. J. . R. Piene (Ed.), Computational Methods for Algebraic Spline Surfaces, 2005, Computational Methods for Algebraic Spline Surfaces, Springer, Oslo, Norway, 2007, pp. 61–79.
URL `http://hal.inria.fr/inria-00132733/en/`