



Université de Montréal

Validité d'un modèle *QuasiNURBS*  
interpolant des données géométriques incertaines

par

Malika Zidani-Boumedien

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Thèse présentée à la faculté des études supérieures  
en vue de l'obtention du grade de  
Philosophiæ Doctor (Ph. D.)  
en informatique

Mars 2006

© Malika Zidani-Boumedien, 2006

Université de Montréal  
Faculté des études supérieures

Cette thèse intitulée

Validité d'un modèle *QuasiNURBS*  
interpolant des données géométriques incertaines

présentée par  
Malika Zidani-Boumedien

a été évaluée par un jury composé des personnes suivantes :

Pierre Poulin  
président-rapporteur

Neil F. Stewart  
directeur de recherche

Victor Ostromoukhov  
membre du jury

Thomas J. Peters  
examineur externe

Pierre Poulin  
représentant du doyen de la FES

# Résumé

Un des problèmes fondamentaux dans l'étude de la robustesse des méthodes numériques en modélisation des solides est que les données fournies à l'algorithme présentent souvent des incohérences géométriques (comme des décalages ou superpositions de surfaces). Ces problèmes sont dus aux erreurs provoquées par :

- une incertitude dans les données de départ,
- l'arithmétique en virgule flottante,
- l'approximation des frontières des faces d'un objet par des courbes de bas degré, par exemple de degré 3.

Notre premier objectif était d'analyser l'erreur dans le contexte des opérations géométriques entre solides soumis à ces trois erreurs. Mais nous nous sommes heurtés à un problème préalable important : que représentent les données incohérentes supposées décrire les objets soumis à la méthode numérique ? Quelle est leur **réalisation**<sup>1</sup> ? Comment s'assurer que les données géométriques concordent avec les propriétés topologiques de l'objet ?

Il est donc primordial de définir rigoureusement les sous-ensembles de  $R^3$  (objets) présentés à l'algorithme. Le but de cette thèse est donc de résoudre ce premier problème d'incohérence et d'incertitude sur les données apportant ainsi une contribution à la solution du problème global de définition de la robustesse des méthodes numériques en modélisation des solides.

---

<sup>1</sup>Dans toute la thèse, les mots écrits en mode **Sans Serif** sont des mots qui nécessitent une définition rigoureuse qui sera donnée plus loin que le paragraphe en question.

Nous proposons [4] une solution basée sur le théorème de Whitney. Ce théorème nous permet de définir des ensembles *QuasiNURBS*<sup>2</sup> qui seront la réalisation des données incohérentes de départ pour chaque objet. Nous illustrons le problème et discutons de la solution proposée sur un exemple. Une approche analogue pour les surfaces subdivisées est aussi décrite [74].

L'introduction dans notre solution des *patches* de surface découpées (*trimmed patches*)<sup>3</sup>, permet de résoudre le problème pour des modèles dont les frontières ne peuvent pas être décrites en utilisant des courbes de bas degré et d'éviter donc la troisième erreur citée ci-dessus.

Dans cette étude, nous nous intéressons donc à la qualité géométrique et topologique de la réalisation en démontrant la **validité** des données auquel cas sera défini l'ensemble *QuasiNURBS* correspondant. Nous énonçons à cet effet, des théorèmes précisant les hypothèses pour que ces ensembles soient bien formés, *i.e.*, qu'ils **correspondent** aux données et que leurs frontières ne s'autointersectent pas. Les critères de la détection de l'autointersection sont discutés et illustrés [5]. La détection d'autointersection est un sujet délicat, qui se complique par le fait que les sous-ensembles de  $R^3$  considérés sont des *trimmed patches* mais dans tous les cas, le test de détection nécessitera de borner la variation des normales. Nous énonçons et démontrons à cet effet un théorème permettant d'obtenir ces bornes.

Nous concluons par une brève description de la qualité numérique, *i.e.*, la **stabilité numérique**, d'un algorithme effectuant une opération entre deux objets (c'est-à-dire entre leur représentation) par le biais de l'**analyse inverse de l'erreur**. Nous pensons que notre définition rigoureuse d'une représentation valide permettra maintenant de faire cette analyse.

**Mots clefs** : incertitude des données, *QuasiNURBS*, Bézier, *trimmed patches*, réalisation, valide, extension de Whitney, constante de Lipschitz, autointersection, vecteur normal.

---

<sup>2</sup>NURBS = Non Uniform Rational B-Spline

<sup>3</sup>Les expressions anglaises seront en italique.

# Abstract

One of the fundamental problems in the study of robustness of numerical methods in solid modelling is that the data provided to the algorithm is often inconsistent (including gaps and overlaps of surfaces). These problems are due to error caused by :

- uncertainty in the input data,
- the use of floating-point arithmetic,
- the approximation of the boundaries of object faces by low degree curves (say, of degree 3).

Our principal goal was to perform error analysis in the context of geometric operations on solids, in the presence of the types of error mentioned above. Immediately, however, we encountered an important preliminary problem : what set is actually represented by the inconsistent data provided to the method ? What is their actual **realisation** ? How can we be sure that they are consistent with the topological data associated with the object ?

Consequently, it is fundamental to define rigorously the subsets of  $R^3$  (the objects) presented to the algorithm. The goal of this thesis is to solve this preliminary problem of inconsistent and uncertain data, thus making a contribution to the solution of the overall problem of defining robustness of numerical methods in solid modelling.

We propose a solution [4] based on the Whitney Extension Theorem. This theorem allows us to define what we call QuasiNURBS<sup>4</sup> sets, which will be the **realisation** of the inconsistent data for each input object. We illustrate the problem

---

<sup>4</sup>NURBS = Non Uniform Rational B-Spline

and discuss the proposed solution in an example. An analogous approach is also proposed for combined subdivision surfaces [74].

The fact that our approach deals with trimmed patches means that we deal with patches for which the boundary cannot be represented by low-degree curves, and thus with the third source of error itemized above.

In this study, we are concerned with the geometric and topological quality of the **realisation** *i.e.*, with well-formedness. In the case of well-formed objects, the data can be considered valid, and there will exist a corresponding QuasiNURBS set. We give theorems which will guarantee various aspects of well-formedness, *i.e.*, that there exist sets with non-selfintersecting boundaries that correspond to the given data. The criteria for non-selfintersection are presented in [5]. In fact, this problem of non-selfintersection is made more complicated by the fact that we are dealing with trimmed patches, but however these difficulties are resolved, it will be necessary to find bounds on the variation of the face normals. We give a theorem which provides such bounds.

We conclude with a brief description of the question of the quality of computed answers, as reflected in the numerical stability of an algorithm to effect an operation on two objects (or, more precisely, their representations) by using a backward error analysis. We think that our rigorous definition of a valid representation will now permit such an analysis.

**Keywords** : Data uncertainty, QuasiNURBS, Bézier, trimmed patches, realisation, well-formed, Whitney extension, Lipschitz constant, selfintersection, normal vector.

# Table des matières

|                                                                                         |           |
|-----------------------------------------------------------------------------------------|-----------|
| Résumé                                                                                  | iv        |
| Abstract                                                                                | vi        |
| Remerciements                                                                           | xiii      |
| <b>1 Introduction</b>                                                                   | <b>1</b>  |
| 1.1 Description du problème . . . . .                                                   | 4         |
| 1.2 Contribution . . . . .                                                              | 6         |
| <b>2 État de l’art</b>                                                                  | <b>8</b>  |
| 2.1 Problème général de robustesse . . . . .                                            | 8         |
| 2.2 Interpolation transfinie . . . . .                                                  | 12        |
| 2.3 Opération de découpage ou <i>trimming</i> . . . . .                                 | 13        |
| 2.4 Représentation et réalisation . . . . .                                             | 15        |
| 2.4.1 Modèles ou réalisations . . . . .                                                 | 15        |
| 2.4.2 Représentations utilisées en pratique . . . . .                                   | 17        |
| 2.5 Détection d’autointersection . . . . .                                              | 20        |
| <b>3 Description de la réalisation</b>                                                  | <b>22</b> |
| 3.1 Ensemble <i>QuasiNURBS</i> . . . . .                                                | 22        |
| 3.1.1 Définition et qualité de la réalisation : l’ensemble <i>Quasi-NURBS</i> . . . . . | 23        |
| 3.1.2 Représentation valide . . . . .                                                   | 28        |
| 3.1.3 Exemple numérique . . . . .                                                       | 29        |



|          |                                                                 |           |
|----------|-----------------------------------------------------------------|-----------|
| 3.2      | Approche pour les surfaces subdivisées . . . . .                | 36        |
| <b>4</b> | <b>Vérification de non-autointersection</b>                     | <b>41</b> |
| 4.1      | Formulation mathématique de la conjecture . . . . .             | 42        |
| 4.1.1    | Hypothèses générales . . . . .                                  | 43        |
| 4.1.2    | Cas d'un domaine simple . . . . .                               | 45        |
| 4.1.3    | Cas d'un domaine composé . . . . .                              | 45        |
| 4.2      | Contre-exemples . . . . .                                       | 46        |
| 4.3      | Énoncé du théorème . . . . .                                    | 57        |
| <b>5</b> | <b>Non-intersection de <i>trimmed patches</i></b>               | <b>60</b> |
| 5.1      | Comment détecter une autointersection sur une surface perturbée | 61        |
| 5.2      | Bornes sur les normales de la <i>patch</i> perturbée . . . . .  | 62        |
| 5.3      | Non-intersection des faces du <i>QuasiNURBS</i> . . . . .       | 69        |
| <b>6</b> | <b>Conclusion</b>                                               | <b>72</b> |
| 6.1      | Conclusion de notre travail . . . . .                           | 72        |
| 6.2      | Travaux futurs . . . . .                                        | 74        |
| 6.2.1    | Principe de l'analyse inverse de l'erreur . . . . .             | 74        |
| 6.2.2    | Problèmes ouverts . . . . .                                     | 76        |
|          | <b>Bibliographie</b>                                            | <b>78</b> |
| <b>A</b> | <b>Régularisation</b>                                           | <b>85</b> |
| A.0.3    | Régularité des ensembles . . . . .                              | 85        |
| A.0.4    | Régularité de l'opération . . . . .                             | 86        |
| <b>B</b> | <b>Définitions liées à la topologie algébrique</b>              | <b>88</b> |

# Table des figures

|     |                                                                                             |    |
|-----|---------------------------------------------------------------------------------------------|----|
| 1.1 | Problème de robustesse en Modélisation des Solides : Intersection de deux solides . . . . . | 2  |
| 1.2 | Problème de robustesse en Géométrie Numérique : Triangulation de Delaunay [67]. . . . .     | 2  |
| 2.1 | Données géométriques . . . . .                                                              | 19 |
| 2.2 | Données topologiques associées à la figure 2.1 . . . . .                                    | 19 |
| 3.1 | Définition de la perturbation sur $\partial D$ . . . . .                                    | 26 |
| 3.2 | Intersection de deux solides simples . . . . .                                              | 30 |
| 3.3 | Effet de l'arithmétique en virgule flottante . . . . .                                      | 31 |
| 3.4 | Pré-image de l'intersection avec le plan $z = 21$ . . . . .                                 | 32 |
| 3.5 | Deux <i>trimmed patches</i> adjacentes obtenues par subdivision combinée                    | 39 |
| 3.6 | Intersections indésirables de <i>patches</i> SCT adjacentes . . . . .                       | 40 |
| 4.1 | Exemples d'autointersection [76] . . . . .                                                  | 42 |
| 4.2 | Surface composée de deux <i>patches</i> qui s'autointersectent et sa projection . . . . .   | 48 |
| 4.3 | Champ des normales de la surface composée de la figure 4.2 . . .                            | 48 |
| 4.4 | Surface composée de trois sous- <i>patches</i> : domaine paramétrique . .                   | 50 |
| 4.5 | Surface composée dans $R^3$ . . . . .                                                       | 50 |
| 4.6 | Superposition de deux <i>patches</i> adjacentes . . . . .                                   | 53 |
| 4.7 | Domaine paramétrique non simplement connexe . . . . .                                       | 55 |
| 4.8 | Autointersection : domaine avec trou . . . . .                                              | 56 |

|     |                                                           |    |
|-----|-----------------------------------------------------------|----|
| 4.9 | Illustrations des conditions $g_1$ et $g_2$ . . . . .     | 58 |
| A.1 | Ensemble non semi-analytique [62]. . . . .                | 86 |
| A.2 | Intersection conventionnelle versus régularisée . . . . . | 87 |
| B.1 | Simplexes utilisés . . . . .                              | 89 |
| B.2 | Collection de simplexes . . . . .                         | 90 |
| B.3 | $k$ -cycles sur un complexe simpliciale. . . . .          | 90 |

*Avec la grâce de Dieu,  
j'ai pu terminer ce travail et apporter mon humble contribution  
à la modélisation des solides. C'était pour moi comme la  
traversée d'un océan avec tempêtes et eaux calmes. Mais comme  
toute épreuve, on sort grandi d'une telle expérience et c'est  
un réel bonheur que d'atteindre un autre rivage.  
Car, là, tout reste à découvrir et la fin d'une épopée  
est le début d'une autre...*

*Et c'est avec plaisir que je dédie ce travail  
À mon précieux mari pour son extraordinaire et sincère soutien,  
À mes enfants chéris qui me donnent la force de persévérer et qui  
ont été patients et sages durant mes absences,  
À ma mère qui a su faire de moi une battante,  
À mon père qui m'a appris la patience et m'a comblée de ses  
invocations.*

*Et je terminerai par cette invocation :  
"Dieu, fais nous profiter de ce que Tu nous as appris et  
apprends nous ce qui nous sera bénéfique et multiplie  
nos connaissances."*

# Remerciements

On m'avait dit, au début de ma thèse, que j'étais entre de bonnes mains et je tiens à le confirmer. Efficacité, compétence, disponibilité, rigueur et ouverture d'esprit sont là quelques unes des qualités que j'ai trouvées en mon professeur que je remercie sincèrement pour m'avoir appuyée scientifiquement et financièrement. Parfois mettant du stress, parfois lâchant du lest, le travail devait avancer régulièrement. Cependant, son côté humain m'a permis de garder un équilibre entre travail et famille et son intuition scientifique m'a éclairée sur plusieurs sujets. Donc, à vous mon prof : Merci pour tout !

Je remercie le professeur Pierre Poulin d'avoir accepté de présider ce jury et surtout d'avoir répondu à mes nombreuses questions sur l'infographie. La pertinence de ses réponses et de ses questions m'ont beaucoup appris sur mon propre sujet et sa sympathie a apporté beaucoup de détente au milieu d'un travail ardu. Je remercie aussi le professeur Victor Ostromoukhov pour avoir accepté d'être membre du jury, ainsi que les professeurs Esma Aïmeur et Farida Chériet pour avoir gentiment accepté de répondre à toutes mes questions sur leur domaine respectif.

Je tiens à exprimer ma reconnaissance au professeur Lars-Erik Andersson qui a accepté ma contribution dans deux de nos papiers soumis. Je remercie spécifiquement et sincèrement le professeur Thomas Peters d'abord d'avoir accepté de faire partie du jury et aussi de nous avoir enrichi par tous ses commentaires très pertinents. Étant un des spécialistes du sujet, sa présence dans le jury m'honore. Je tiens aussi à exprimer toute ma reconnaissance à Mr. J. Bisceglia pour avoir gracieusement

accepté de faire pour nous des calculs avec la librairie DTNURBS, son aide nous a été précieuse.

Finalement, que tous ceux et celles qui m’ont encouragé et fait des invocations pour moi, parmi mes chers parents, mes chers frères et soeurs, parmi ceux de ma grande famille, toutes mes fidèles amies où qu’elles soient, trouvent ici autant et même plus de gratitude et profonde sympathie que leur soutien et encouragements. Mes remerciements vont aussi à mes collègues du Ligum et des autres laboratoires qui ont gentiment accepté de m’aider quand j’avais des problèmes et en particulier à ma sympathique “coéquipière” de notre laboratoire de modélisation, ainsi qu’à toutes les employées de l’administration ou de la bibliothèque qui m’ont toujours bien accueillie et qui sont très efficaces dans leur travail.

Un tel travail, de longue haleine, n’a pu aboutir que grâce à l’implication, d’une manière ou d’une autre, de toutes ces personnes.

# Chapitre 1

## Introduction

*“Le fini ne se distingue de l’infini que par l’imperfection”*

*Reverdy Pierre, poète français, 1889-1960.*

Une des plus importantes préoccupations aujourd’hui, en modélisation des solides, est la non-robustesse dans les opérations géométriques sur les objets. Quelques exemples des conséquences de la non-robustesse des algorithmes aussi bien géométriques que numériques prouveront la nécessité de cette étude :

- un algorithme d’intersection de deux solides [4, 34], mis en échec dans le cas de deux polyèdres tangents intérieurement à cause du décalage du point d’intersection (à cause des erreurs d’arrondis) (voir FIG. 1.1),
- un exemple d’une triangulation de Delaunay faussée par des erreurs de précision, [65] (FIG. 1.2),
- et finalement un exemple plus spectaculaire de non-robustesse d’un algorithme numérique cette fois : l’explosion de la fusée Ariane en juin 1996 (débordement numérique).

On voit donc que les conséquences de la non robustesse peuvent être désastreuses dans différents domaines. D’autres exemples d’incohérences sont donnés dans [40, 53]. En fait, plusieurs théorèmes géométriques classiques, vrais dans l’espace Euclidien infini, sont mis en échec dès qu’on passe en mode fini, monde de la machine.

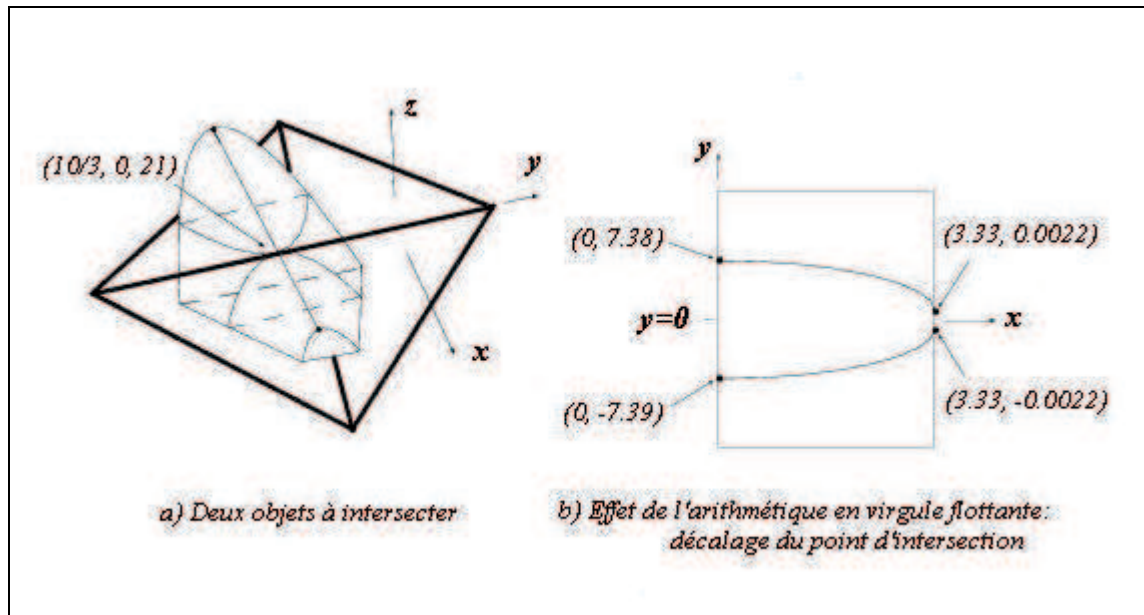


FIG. 1.1: Problème de robustesse en Modélisation des Solides : Intersection de deux solides

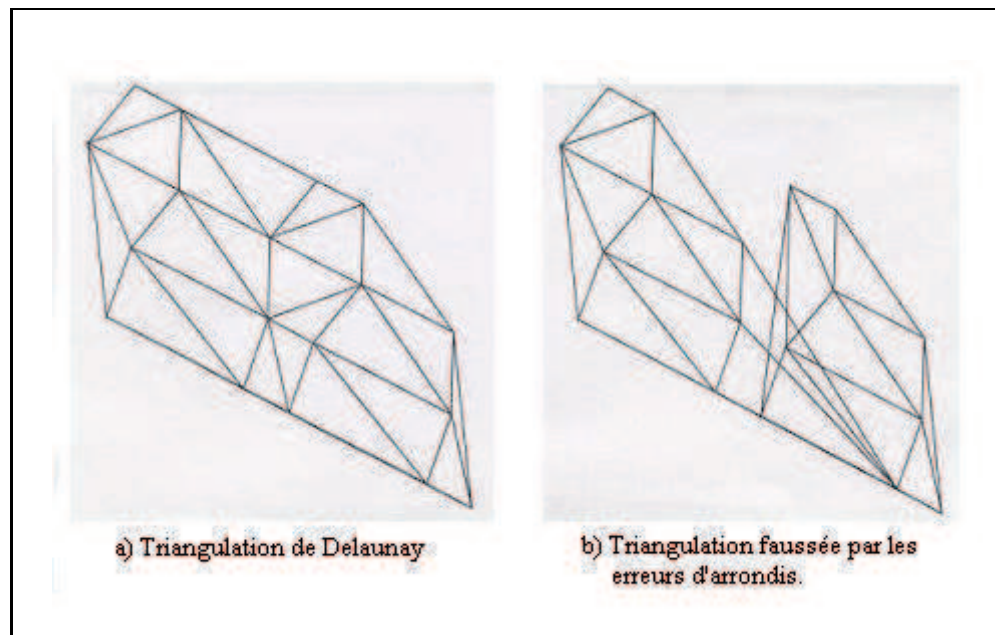


FIG. 1.2: Problème de robustesse en Géométrie Numérique : Triangulation de Delaunay [67].



Revenons au premier exemple pour décrire en détail les conséquences désastreuses dans le cadre de la modélisation des solides. Le décalage du point d'intersection, qui peut sembler insignifiant, va provoquer un désordre dans la topologie de l'objet résultant. En effet, dans ce cas d'intersection, un segment de droite (l'axe de symétrie de la face courbe) d'un des solides intersecte la face supérieure (perpendiculaire à l'axe  $z$ ) du tétraèdre en un point extérieur à cette face alors que ce même segment intersecte la face inférieure (perpendiculaire à l'axe  $x$ ) en un point situé sur l'arête commune aux deux faces donc appartenant à la face supérieure. À cause de l'arithmétique en virgule flottante, un segment de droite coupe donc un plan en deux points. Dans notre cas de figure, il y a donc un décalage entre deux faces de l'objet résultant de l'intersection. Dans une autre situation [34, FIG. 1], l'intersection du segment de droite avec ces deux faces donnerait deux points l'un intérieur et l'autre au bord de la face supérieure, ce qui provoquera un écart entre les deux faces résultantes. Dans tous les cas, l'algorithme d'intersection va bloquer au moment de reconstituer l'objet final [34].

L'étude de la robustesse des algorithmes géométriques a pris son essor vers la fin des années 80 [70, 32, 31, 54, 22, 19] et continue jusqu'à ce jour de faire couler beaucoup d'encre [34, 78, 71]. En fait, une vingtaine d'années après, Farouki [24], un des spécialistes du domaine, observe qu'aucune théorie satisfaisante permettant de décrire comment approcher, de manière cohérente, les côtés et surfaces du solide représenté, n'a réussi à se mettre en place. Plusieurs définitions de la robustesse sont utilisées de manière informelle. Intuitivement, un algorithme géométrique est non robuste s'il produit un résultat qui présente des incohérences géométriques comme des décalages ou des superpositions entre des surfaces supposées contiguës. En fait, Hoffmann [33] souligne que l'origine de la non-robustesse est l'interaction entre données numériques approximatives et données symboliques exactes. Ainsi, l'un des buts de notre travail est de donner une définition rigoureuse et utile à ce qu'on appelle "robustesse".

## 1.1 Description du problème

Pour effectuer une opération géométrique sur un solide (ou entre solides), ce dernier est soumis à l'algorithme sous forme de représentation. La représentation est un ensemble de données numériques (plus précisément géométriques) et topologiques décrivant un objet, de manière cohérente ou pas.

Le problème de robustesse numérique (ou simplement robustesse), résulte de l'incertitude sur les données représentant l'objet, de l'utilisation de l'arithmétique en virgule flottante et de l'approximation par des courbes de bas degré. L'analyse inverse de l'erreur serait alors une méthode pour vérifier le conditionnement d'un problème géométrique et la stabilité numérique de l'algorithme.

En effet, nous pouvons espérer montrer que le résultat de l'algorithme est exactement la solution d'un problème géométrique proche mais légèrement perturbé par rapport au problème géométrique original [35]. Et la relation entre ces deux problèmes est habituellement ignorée [69].

D'autre part, la représentation d'un objet, que ce soit en entrée ou en sortie, doit être **valide**. En ce sens que les représentations doivent rigoureusement définir un ensemble fermé régulier (voir annexe), *i.e.*, un  $r$ -ensemble [61, 63]. Avant d'entamer toute analyse, il est donc primordial de donner un sens aux représentations *i.e.*, de définir rigoureusement l'objet représenté par ces données. En plus, la représentation comprend des entités géométriques souvent obtenues par des opérations géométriques, par exemple intersection entre deux surfaces. Il s'ensuit que l'objet est composé de *patches* de surface découpées (*trimmed patches*).

Considérons le cas de deux faces adjacentes décrites par des données géométriques et topologiques. Les **données géométriques** sont :

1. dans  $R^3$ ,
  - deux *patches* de NURBS  $\mathbf{F}[D]$  et  $\mathbf{F}'[D']$  représentées par leurs points de contrôle, décrivant les faces de l'objet et avec deux possibilités pour  $D$  :  
 cas 1 :  $D = [0, 1]^2 (= D')$  ; dans ce cas,  $\mathbf{F}[D]$  et  $\mathbf{F}'[D']$  sont des *patches*,  
 cas 2 :  $D$  et  $D'$  sont inclus dans  $[0, 1]^2$ , alors  $\mathbf{F}[D]$  et  $\mathbf{F}'[D']$  sont des

*trimmed-patches* (c'est le cas que nous considèrerons dans la suite du travail),

- éventuellement la courbe commune aux deux faces (notons-la  $\mathbf{b}(t)$ ), représentée par des points de contrôle,
- et éventuellement aussi les extrémités  $\mathbf{v}_1$  et  $\mathbf{v}_2$  de la courbe commune aux deux faces,

2. dans  $R^2$ ,

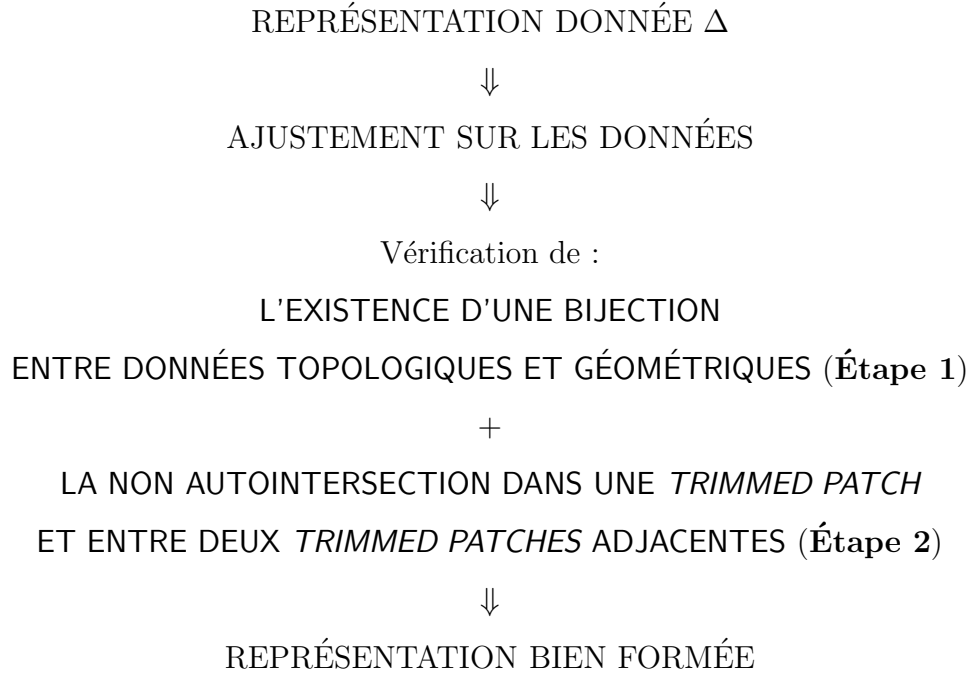
les pré-images des frontières de  $\mathbf{F}[D]$  et  $\mathbf{F}'[D']$ , c'est-à-dire les  $p$ -courbes  $\mathbf{p}(t)$  et  $\mathbf{p}'(t)$ , décrivant une courbe fermée dans  $R^2$  limitant les domaines  $D$  et  $D'$  respectivement. Dans le cas 1, ces courbes sont les bords du carré  $[0, 1]^2$ . On considère que les  $p$ -courbes sont aussi des courbes de NURBS.

Quant aux **données topologiques**, elles consistent en la donnée d'un complexe cellulaire, ici formé de deux faces et d'une arête.

Les données topologiques et géométriques sont souvent incohérentes. Comme nous considérons que les données topologiques sont exactes, il devient nécessaire de perturber les données géométriques (NURBS) afin de donner un sens à la représentation. Les faces perturbées seront alors les frontières du r-ensemble : l'ensemble *QuasiNURBS*. Cet ensemble *QuasiNURBS* de  $R^3$  est la **réalisation** associée à la représentation. Ainsi, si une représentation, notée  $\Delta$ , est **bien formée**, alors il existera une et une seule réalisation associée à cette représentation. Et on dira qu'une représentation est bien formée si :

- il existe une **correspondance exacte** entre la structure géométrique et la structure topologique données,
- les données géométriques ajustées (pour obtenir la correspondance exacte) ne s'autointersectent pas.

Le schéma suivant permet de résumer ce qui précède :



Ce schéma décrit les étapes nécessaires pour obtenir une représentation bien formée ou valide partant d'une représentation donnée tout à fait aléatoire.

## 1.2 Contribution

Une fois démontrée la validité de la représentation, c'est-à-dire, une fois les étapes 1 et 2 du schéma précédent (sec.1.1) vérifiées, le calcul de la distance entre les objets décrits par ces représentations (moyennant le choix d'une métrique), aura un sens et permettra de faire une analyse de l'erreur liée au problème dont les données géométriques sont incertaines.

Notre travail a donc pour but, non pas de concevoir une nouvelle méthode "robuste" pour faire une opération géométrique mais de contribuer à l'élaboration d'une méthodologie rigoureuse pour mesurer le degré de robustesse d'un algorithme quelconque dans un contexte précis : le cas de données de départ incertaines et d'utilisation de l'arithmétique en virgule flottante, l'arithmétique en virgule flottante étant largement utilisée dans les calculs numériques. Nous ne prétendons pas résoudre totalement le problème de robustesse mais nous pensons avoir posé

les bases en définissant rigoureusement les éléments qui permettront d'établir une théorie consistante sur l'analyse de l'erreur dans les opérations géométriques entre solides, évitant ainsi de traiter, cas par cas et de manière heuristique, les résultats erronés dues aux erreurs citées ci-dessus.

Une des difficultés dans le cas général, est que les faces du solide considéré sont des *trimmed patches* quelconques, limitées par plusieurs frontières courbes et non des *patches* usuelles, *i.e.*, des *patches* dont le domaine paramétrique est un pavé ou un triangle de  $R^2$ . Et bien que ce soit un cas très fréquent [1, 73], il n'est toujours pas maîtrisé et continue de susciter beaucoup d'intérêt. Cette particularité est prise en compte dans notre étude.

Ainsi, nous présenterons au chapitre suivant une synthèse des travaux liés à notre sujet. Dans le chapitre 3.1, nous nous intéresserons à la première étape du schéma de la section 1.1 soit l'existence d'une bijection entre données topologiques et géométriques ainsi que la définition d'une représentation bien formée ou valide. Ce chapitre comprendra aussi une application aux surfaces de subdivision. L'étape 2 de ce schéma, soit la non-autointersection, sera développée dans deux chapitres : Le chapitre 4 sera une présentation du critère de Volino-Thalmann, énoncé d'une manière rigoureuse ainsi que des exemples et contre-exemples liés à ce critère. Le chapitre 5 sera l'application de ce critère au cas particulier des *trimmed patches* perturbées. On y fournit un résultat permettant de borner les vecteurs normaux. Ce résultat pourra être utilisé pour la formulation de théorème éliminant les intersections parasites entre surfaces adjacentes. On conclura au chapitre 6 en soulignant l'importance de notre contribution et les travaux qui peuvent s'ensuivre particulièrement l'étude de la stabilité numérique par l'analyse inverse de l'erreur.

# Chapitre 2

## État de l'art

Cette thèse est une contribution au problème général de robustesse donc la première section de ce chapitre sera une description de la littérature parue sur ce sujet. Ce problème général fait appel à plusieurs concepts, opérations ou méthodes qui ont fait l'objet, à part entière, d'une dense bibliographie. Ce chapitre sera donc une brève revue des travaux liés à notre étude. Nous présenterons, en particulier, dans la section 2.2, les principaux travaux sur les méthodes d'interpolation transfinie, dans la section 2.3, ceux concernant l'opération de découpage, viendra ensuite la section 2.4 sur la définition de représentation et réalisation et enfin la dernière section sur les méthodes d'autointersection.

### 2.1 Problème général de robustesse

Il existe déjà des approches pour un calcul géométrique robuste qui traitent, d'une manière succincte, le cas des données incertaines. Parmi les solutions proposées pour rendre les algorithmes géométriques “plus robustes” , nous citerons :

- les *heuristiques* des  $\varepsilon$  ; celles-ci consistent à se donner une tolérance  $\varepsilon$  et décider que ce qui est proche de zéro (à  $\varepsilon$  près) est égal à zéro autrement dit que  $x=y$  si  $|x - y| < \varepsilon$ ,
- l'arithmétique des intervalles qui consiste à remplacer un nombre par un intervalle centré en ce point,

- l'élimination des cas dégénérés (ou cas limites) en introduisant des perturbations,
- la modification des données symboliques,
- différents schémas basés sur une extension de la précision arithmétique pour représenter les nombres de manière exacte (ou, adaptativement [58] avec une précision arbitrairement grande).

En fait, selon Lieutier [48], “dans l'industrie, la réalisation et la maintenance d'algorithmes géométriques comportant une part combinatoire (topologique) et une part numérique (géométrique) sont très coûteuses. La réalisation de tels algorithmes ne procède pas d'une méthodologie rigoureuse et reproductible mais d'un savoir-faire personnel difficilement transmissible. Si les conséquences semblent mineures et n'affecter que des cas limites peu fréquents, en fait, nous remarquons que la gestion des  $\varepsilon$  et des cas limites devient la partie la plus coûteuse en mise au point et en maintenance”. Ceci est dû au fait que ces solutions sont proposées sans que le problème ne soit bien défini. C'est ce qui est arrivé historiquement et c'est en partie pour cela que le problème persiste. Voilà la lacune principale qui a suscité notre intérêt.

Les méthodes décrites ci-dessus, et bien d'autres, essaient d'éviter ou de rattrapper des calculs faussés par les erreurs de conversion ou dues à la méthode numérique d'approximation utilisée [54, 65], calculs qui mèneront à des décisions incohérentes, provoquant des erreurs graves, surtout quand les tests ne sont pas indépendants, menant à l'échec des méthodes numériques classiques, théoriquement correctes et aussi à l'échec des algorithmes de maillage [24].

Ces méthodes apportent toutes une contribution au problème posé et il n'est pas exclu qu'on utilise l'une ou l'autre de ces méthodes localement. Cependant, elles présentent aussi certaines limitations qui laissent encore ouvert le problème de robustesse des opérations entre solides et que nous résumerons dans ce qui suit :

- il y a des problèmes graves avec la méthode des heuristiques des  $\varepsilon$  pour les représentations par frontières (B-rep). Les  $\varepsilon$  évitent certaines incohérences

mais en provoquant d'autres [4, 34] ; par exemple, la notion de colinéarité n'est plus transitive. En effet, nous pouvons facilement avoir une polyligne où tous les points sont colinéaires trois par trois à  $\varepsilon$  près, sans que la polyligne finale ne soit une droite. Dans chaque triplet de points colinéaires, une petite perturbation inférieure à  $\varepsilon$ , en a fait des points colinéaires mais il n'y a pas de petite perturbation qui permettra d'avoir une polyligne finale droite. Par suite, si nous appliquons une petite perturbation inférieure à  $\varepsilon$  sur chacune des données, la taille de la perturbation globale n'est en rien assurée si le nombre de segments formant la polyligne est important. En général, la notion de transitivité de l'égalité est perdue [65].

- l'arithmétique des intervalles est une méthode lente et pour le moment, encore pessimiste : au lieu de manipuler des nombres, nous manipulons des intervalles dont nous ne contrôlons pas toujours la taille à cause de l'utilisation de l'arithmétique à virgule flottante et de l'effet enveloppant (ou *wrapping effect*). L'effet enveloppant est dû au fait que l'image d'un intervalle vectoriel (par exemple, un rectangle) n'est pas un intervalle vectoriel. La solution intervalle sera alors, par exemple, un rectangle contenant (ou enveloppant) l'image [55, 44]. Et le problème n'est toujours pas réglé, pour les mêmes raisons que la méthode des heuristiques décrite ci-dessus : nous ne connaissons pas la forme topologique de l'intersection à l'intérieur de la zone d'intersection.
- le traitement des cas dégénérés, comme nous l'avons précisé plus haut, est très couteux en pratique [48]. Ces méthodes ont pour but d'éliminer les tests du type “un point est-il sur une ligne ? le résultat est-il égal à zéro ?” pour éviter les cas limites qui provoquent les dégénérescences. D'une part, l'élimination de ces “positions dégénérées” [32] se fait en perturbant la position des éléments les uns par rapport aux autres et rien ne prouve que l'intégrité de l'objet géométrique présenté au problème, soit maintenue. D'autre part, l'élimination automatique des cas limites n'est pas toujours souhaitable en modélisation des solides. Pour la méthode SOS (*Simulation*



*Of Simplicity*), Edelsbrunner et Mücke [21] prouvent qu'on peut éliminer les cas dégénérés en perturbant les données dans des cas simples. Ils ne considèrent que des opérations géométriques primitives, une opération primitive étant une fonction  $f$  qui associe à un ensemble  $Q$  de  $k$  objets les valeurs  $+1$ ,  $0$  et  $-1$  où  $0$  est le cas dégénéré. Cette méthode nécessite de savoir quand  $f(Q) = 0$ , c'est à dire qu'elle nécessite l'utilisation de l'arithmétique exacte. En fait, ils font une perturbation symbolique, en remplaçant toutes les données numériques par un polynôme en  $\varepsilon$ . Cette technique théorique permet d'élaborer des méthodes sûres et qui permettent à l'utilisateur de travailler directement avec les données originales sans se soucier du calcul du polynôme en  $\varepsilon$ . Cependant, trouver le bon polynôme n'est pas trivial même dans des cas simples.

- quant à l'augmentation de la précision, cette méthode, d'une part, ne convient pas si les données sont incertaines et d'autre part, est très coûteuse. En effet, les données peuvent être obtenues grâce à des capteurs sensoriels ou être le résultat d'un autre programme et être, par conséquent, imprécises. Est-il, alors, vraiment pertinent de représenter les nombres de manière exacte, méthode coûteuse en temps calcul et en espace mémoire, pour donner des résultats précis alors que les données sont incertaines [35] ?

C'est pour cela que bien poser le problème de robustesse est crucial ; par exemple, certains chercheurs proposent l'utilisation de l'arithmétique exacte pour régler le problème. Ceci est ou n'est pas justifié selon le problème posé :

- si le problème est défini par des données exactes, c'est une solution très chère mais fructueuse [34, Sec. 3.2],
- si le problème est défini par des données incertaines, c'est non seulement une solution très chère mais en plus, inutile.

Un résumé des méthodes décrites ci-dessus est donné dans [32].

Ces travaux soulignent bien la nécessité de poser le problème de manière rigoureuse en commençant par définir clairement les opérateurs entrant dans le processus géométrique (ou en sortant) pour éviter d'utiliser des méthodes coûteuses

et infructueuses.

## 2.2 Interpolation transfinie

Les méthodes d'interpolation transfinie se basent sur des données continues : un ensemble de courbes à interpoler au lieu d'un ensemble de points.

La première publication à ce sujet a été le travail de S. A. Coons [11]. Il a défini un interpolant transfini qui interpole les quatre courbes frontières d'une *patch* et où les fonctions d'interpolation sont des polynômes, linéaires pour une continuité  $C^0$  et d'Hermite pour une continuité  $C^1$ . Gordon [26] a observé que la *patch* de Coons peut en fait s'écrire comme somme booléenne de projecteurs à une dimension. Il a ensuite généralisé ce concept en construisant une surface d'interpolation transfinie passant par un réseau de courbes parallèles .

Cette classe de surfaces d'interpolation transfinie est en fait une solution à un problème variationnel : la minimisation de fonctionnelles liées à la régularité de la surface d'interpolation. Par exemple, les *patches* de Coons minimisent la torsion [23].

Les *patches* de Coons et les surfaces de Gordon sont applicables sur des domaines rectangulaires ou triangulaires. Pour des domaines limités par un cercle ou un polygone, simplement connexe ou pas, Gross et Farin [30] ont proposé une méthode d'interpolation transfinie basée sur l'interpolant de Sibson qui utilise le diagramme de Voronoi dans le plan pour interpoler un ensemble de points. Ils proposent de calculer le diagramme de Voronoi pour des données continues, soit des segments de droite ou des courbes au lieu des points. Leur méthode résoud le problème de trouver une fonction “raisonnablement” lisse [30, p.38], qui est définie sur un domaine fermé et qui interpole exactement la courbe frontière donnée.

Les méthodes exposées plus haut sont des méthodes de calcul qui supposent que les différentes entités géométriques sont certaines et sans ambiguïté, mais en réalité, c'est rarement le cas : les données sont très souvent incohérentes. Et, dans notre contexte, ce qui nous intéresse, n'est pas de construire un interpolant

transfini en tant que tel mais de définir grâce à ce type d'interpolation, un ensemble bien formé ou valide à partir de données incohérentes dans le cadre de l'étude de la robustesse [4, 35]. Nous verrons plus loin (section 3.1.1) le type d'interpolation transfinie qui “réalise” nos objectifs.

## 2.3 Opération de découpage ou *trimming*

Le *trimming* est une opération qui est maintenant dans les normes standard des systèmes d'automatisation industrielle [1, 73]. C'est une opération importante en modélisation géométrique en général et des solides en particulier [28, 36, 45, 57]. Une méthode de *trimming*, en général, consiste à donner une formulation mathématique approximant la partie à découper dans une surface donnée.

Le *trimming* rectangulaire est déjà standard [73]. Une *trimmed* surface rectangulaire est une surface limitée par des courbes dont les pré-images dans l'espace paramétrique sont des lignes isoparamétriques, c'est-à-dire des segments de droite  $(u_1, v)$ ,  $(u_2, v)$ ,  $(u, v_1)$  et  $(u, v_2)$  où  $u_i (i = 1, 2)$  et  $v_i (i = 1, 2)$  sont des constantes.

Dans certaines applications, il est possible d'avoir un domaine paramétrique (associé à une *trimmed patch*) de forme rectangulaire ou triangulaire mais, en modélisation des solides, les objets sont souvent construits par des opérations booléennes [9] par suite le domaine peut être tout à fait quelconque. Une *trimmed* surface NURBS quelconque est dans ce cas moins triviale à déterminer vu que les pré-images de ses frontières ne sont plus des segments de droite dans l'espace paramétrique  $u - v$  mais des courbes. Dans [9], Casale et Bobrow définissent une *trimmed patch* comme une application injective d'un domaine non nécessairement rectangulaire ou triangulaire, borné, régulier (Voir définition en annexe A) de  $R^2$  dans  $R^3$  puis présentent les bases mathématiques d'une opération entre solides dont les frontières sont des *trimmed patches*.

Une approche courante à la construction d'une *trimmed patch* [71, p.1] est de remplacer les surfaces à découper par des approximations polyédrales. Cependant, cette approche ne satisfait pas la précision exigée par certaines applications en

engineering nécessitant des performances aérodynamiques, hydrodynamiques ou autres.

Une idée importante qui reviendra souvent dans cette thèse, est la difficulté de joindre deux *trimmed patches* de ce type, de manière exacte pour former un solide valide. En effet, quand des solides sont construits à partir d'opérations booléennes, par exemple, il est nécessaire de couper les *patches* le long des courbes d'intersection. Et il est, en général, impossible de représenter exactement l'intersection qui est une courbe quelconque non nécessairement une spline de bas degré. La courbe d'intersection formant la frontière de la *trimmed patch* est alors calculée approximativement et ce calcul approximatif provoque souvent des décalages ou chevauchements. En supposant que la pré-image dans le domaine paramétrique de la courbe d'intersection est approché de manière "suffisamment juste", Song et ses coauteurs [71] proposent alors de perturber la surface au voisinage de la courbe d'intersection, en perturbant ses points de contrôle, de manière à faire coïncider les différentes courbes censées représenter la courbe d'intersection et par suite éliminer l'écart ou le chevauchement entre les deux surfaces considérées.

Litke, Levin et Schröder [50] considèrent le *trimming* dans le cas des surfaces de subdivision. Leur méthode pour le *trimming* se base sur le schéma de subdivision combinée [47]. Ce schéma garantit l'interpolation exacte d'une courbe donnée qui formera une frontière de la *trimmed patch*. La subdivision combinée permet la construction de surfaces de subdivision ayant des courbes frontières arbitraires en tant que ces courbes sont  $C^1$ -continues par morceau, paramétrées et possèdent une procédure d'évaluation. Leur méthode est basée sur le schéma de subdivision de Loop. La subdivision, d'une étape à une autre, ajuste le maillage au voisinage de la courbe donnée de manière à rattacher celle-ci à la *patch*. Pour ce faire, ils prennent des points sur la courbe donnée et modifient le stencil de subdivision au voisinage des frontières en considérant ces points comme nouveaux sommets du maillage. Il y aura donc interpolation de la courbe donnée mais la *trimmed patch* ne sera pas égal à la surface originale au voisinage de cette courbe bien qu'on puisse l'approximer à une tolérance près. Selon les auteurs, cette méthode de *trimming*

pour des surfaces subdivisées garantit une interpolation exacte de la courbe de *trimming* donnée et si deux surfaces partagent une même courbe frontière dans l'espace réel (par exemple, leur intersection), les deux surfaces vont concorder exactement en cette courbe et les incohérences topologiques seront évitées. Mais si la courbe, elle-même, est obtenue par calcul, est-il vraiment garanti que ces incohérences seront évitées vu que le calcul se fait en précision finie et en nombre fini d'étapes du processus de subdivision ?

## 2.4 Représentation et réalisation

On a vu dans l'introduction que la probabilité d'avoir des données numériques incohérentes est grande vu que ces données sont, entre autres, le résultat d'un calcul en arithmétique flottante. Des définitions formelles ont été données pour tenter de traduire ce qu'est la représentation discrète de l'objet géométrique dans l'ordinateur et ont donc fait l'objet de plusieurs travaux. Plusieurs de ces définitions sont données à partir d'exemples concrets dans le plan. Elles sont résumées dans [75].

### 2.4.1 Modèles ou réalisations

Hoffmann est l'un des premiers à avoir posé les bases de cette étude [31, 32, 33]. Ces bases ont été énoncées de manière similaire dans [37, 70]. Nous donnerons donc sa description de représentation et de modèle.

Une “représentation discrète” dans l'ordinateur ou simplement une “représentation” d'un objet est une structure de données destinée à décrire l'objet avec des données arithmétiques possiblement imprécises. La représentation contient une liste des données géométriques et logiques. Les données géométriques sont des données numériques et algébriques précisant la position et l'orientation de l'objet comme par exemple les coefficients de l'équation d'un plan pour une face plane. Les données topologiques sont un ensemble d'informations sur la relation entre les différents éléments géométriques comme l'adjacence entre faces, la coïncidence

entre deux sommets, etc.

Une représentation a un modèle, s'il existe un objet dans l'espace euclidien vérifiant les données topologiques exactement et les données géométriques dans le cas où celles-ci sont supposées exactes. Le terme “modèle” est utilisé par Hoffmann mais le terme “réalisation” est utilisé par Hopcroft et Kahn dans [37]. La différence de terminologie n'est pas importante. Notre seul but est de montrer la nécessité de faire la distinction entre la représentation et le modèle (ou la réalisation). Dans notre travail, nous utiliserons le terme réalisation.

Selon Hoffmann, dans le cas où les données sont exactes (précision infinie mais surtout cohérence entre données géométriques et topologiques), la représentation est son propre modèle ; nous appellerons ce modèle un modèle naturel [33]. En fait, l'algorithme manipule le modèle à travers sa représentation mais celui-ci n'est jamais disponible explicitement.

Comme les données numériques dans la représentation sont une approximation des données exactes, la définition suivante qu'il donne, pour comparer entre les données numériques exactes et les données numériques approchées, permet de faire un premier pas dans l'étude de la robustesse.

**Définition 2.4.1.** *Un modèle  $M$  d'une représentation  $R$  est proche à  $\varepsilon$  près si le plus grand écart entre les données numériques de la représentation et ceux exactes du modèle est inférieur ou égal à  $\varepsilon$ .*

Il est important ici de marquer un arrêt sur cette définition. En effet, elle met en évidence la nécessité d'étudier l'écart entre le problème exact et le problème approché mais elle nous permet aussi de mettre en évidence l'importance de notre étude :

- Que veut dire exactement la phrase “écart entre les données numériques de la représentation et ceux exactes du modèle” ?
- Sommes-nous réellement en mesure de déterminer si le modèle existe, c'est-à-dire que la représentation est valide ?

Elle laisse donc ouverte la question de la validité de la représentation car elle ne tient pas du tout compte de l'incohérence entre les données.

La deuxième définition donnée concerne la qualité d'un algorithme pour une opération géométrique binaire.

**Définition 2.4.2.** *Une opération géométrique binaire, par exemple l'intersection, est dite correcte [31] si, pour les représentations  $R_1$  et  $R_2$  présentées au problème, l'opération donne une représentation  $R_3$  telle qu'il existe des modèles  $M_1$ ,  $M_2$  et  $M_3$  où :*

1.  $M_i$  est un modèle de  $R_i$ ,
2.  $M_3 = M_1 \cap M_2$ .

Une définition de la qualité de l'algorithme pour une opération géométrique n-aire est donnée dans [32, 33].

Hoffmann a donc posé le problème à travers l'énoncé de ce que serait une opération correcte mais nous, nous allons plus loin [4, 17] en définissant ce que veut dire “est un modèle de” et en spécifiant rigoureusement comment faire pour vérifier l'existence d'un modèle pour une représentation donnée donnant ainsi un sens à l'égalité  $M_3 = M_1 \cap M_2$ .

La définition précise d'une réalisation a été posée pour la première fois par Desaulniers et Stewart dans [17, 18, 19] dans le cas de faces planaires. En effet, les auteurs introduisent les r-sets quasi-rectilinéaires [17]. Ils présentent en particulier comment les r-sets quasi-rectilinéaires permettent de donner une signification précise à un objet simple, le cube, obtenu par une opération particulière, la translation, en présence d'erreurs d'arrondis. Cette thèse fait donc suite à ce travail dans le sens où nous généralisons cette méthode d'interpolation à des surfaces quelconques non nécessairement linéaires. Nous traitons ensuite les difficultés liées à la perturbation des normales qui font suite à la perturbation des faces géométriques données. Et nous proposerons enfin une définition d'un algorithme robuste.

## 2.4.2 Représentations utilisées en pratique

La majorité des modeleurs géométriques (ACIS [1], STEP [73]) utilisent la représentation décrite précédemment, c'est-à-dire par les données géométriques et

topologiques.

La géométrie de l'objet fait référence aux entités géométriques suivantes : point, courbe, surface et  $p$ -courbe (*pcurves*) ; les  $p$ -courbes sont des courbes paramétriques dans un domaine de  $R^2$ . La géométrie est indépendante de la description topologique de l'objet. La topologie de l'objet, quant à elle, fait référence aux relations ou connexions entre les différentes entités de la description géométrique de l'objet.

Dans la représentation par les bords (ou B-rep), la structure topologique d'un modèle contient les éléments topologiques de base suivants : sommet (*vertex*), arête (*edge*) et face (*face*). Notons  $S$  le solide exact (que nous allons définir plus loin) impliqué dans une opération géométrique telle que l'intersection et dont la représentation incohérente, comprenant les données géométriques et topologiques, est  $\Delta$ .

Les données géométriques dans la représentation  $\Delta$  sont des sous-*patches* *NURBS* (ou *trimmed NURBS patches*) décrivant les faces de l'objet ainsi que les courbes frontières et sommets correspondants. Considérons deux sous-*patches*, notons les  $\mathbf{F}$  et  $\mathbf{F}'$ , définies sur  $D$  et  $D'$  respectivement. La représentation de la courbe d'intersection des deux faces  $\mathbf{F}$  et  $\mathbf{F}'$  comprend deux pré-images, définies par les  $p$ -courbes  $\mathbf{p}$  et  $\mathbf{p}'$  dont l'image est dans  $D$  et  $D'$  respectivement. En plus, la courbe d'intersection est explicitement représentée par une courbe paramétrique  $\mathbf{b}(t)$  à valeurs dans  $R^3$  et qui est proche de  $\mathbf{F}(\mathbf{p}(t))$  et  $\mathbf{F}'(\mathbf{p}(t))$ . Finalement, il y a une représentation explicite de chaque extrémité  $\mathbf{v}_i \in R^3$  ( $i = 1, 2$ ) de la courbe paramétrique  $\mathbf{b}$  (voir FIG.2.1). Dans la pratique,  $\mathbf{v}_1$  et  $\mathbf{v}_2$  sont souvent identiques aux extrémités de  $\mathbf{b}$ .

Les données topologiques (FIG.2.2) forment un 2-cycle topologique [61]. Un 2-cycle correspond à une surface fermée mais non nécessairement une variété topologique (*manifold*) ; les définitions détaillées sont en annexe. Pour illustrer ce qu'est un 2-cycle qui n'est pas un manifold, on pourrait penser à la surface de deux tétraèdres qui ont un sommet ou une arête commune. À chaque 2-*cell* dans les données topologiques correspond, dans les données géométriques de



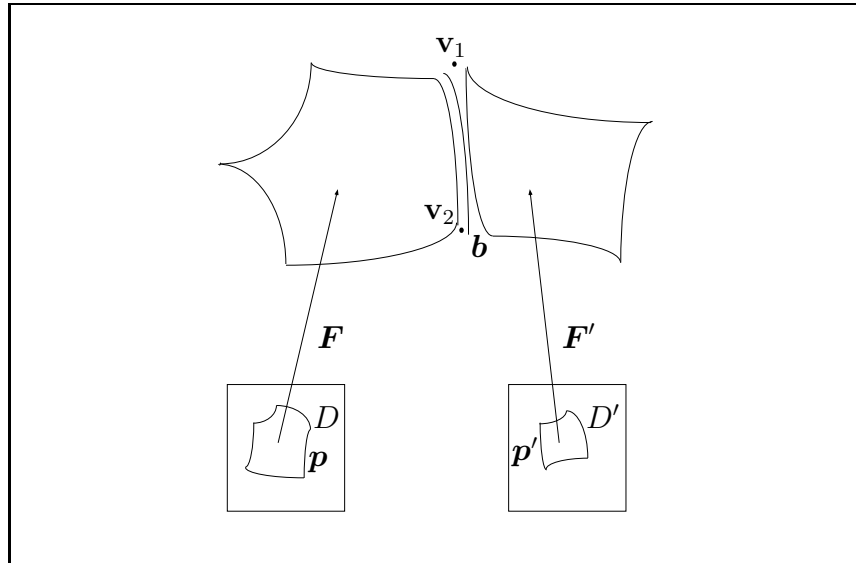
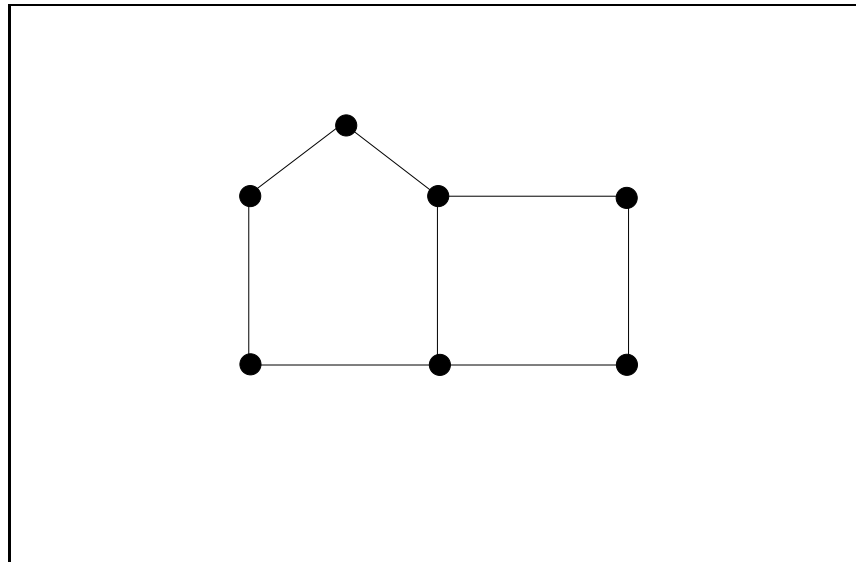
FIG. 2.1: Données géométriques pour deux *trimmed NURBS patches* adjacentes

FIG. 2.2: Données topologiques associées à la figure 2.1

l'utilisateur, une sous-*patch*  $\mathbf{F}[D]$  de  $R^3$ . À chaque 1-*cell* correspond une courbe frontière explicite  $\mathbf{b}$  entre les deux faces adjacentes. Cette dernière courbe n'est pas nécessairement confondue avec l'image d'une des deux  $p$ -courbes associée à chacune des deux faces adjacentes. Finalement, à chaque 0-*cell* dans les données symboliques correspond un point explicite  $\mathbf{v}_i$  dans les données géométriques. Ce point ne coïncide pas, en général, avec l'extrémité des courbes avoisinantes telles que  $\mathbf{b}$ ,  $\mathbf{F}[\mathbf{p}]$  ou  $\mathbf{F}'[\mathbf{p}']$ .

## 2.5 Détection d'autointersection

Plusieurs méthodes ont été présentées dans la littérature. Certaines utilisent les points de contrôle [2] et d'autres les variations du vecteur normal pour connaître la géométrie de la surface [5, 29, 43, 68, 76].

Dans [2], la détection d'autointersection se fait en utilisant une condition suffisante sur les points de contrôle d'une *patch* triangulaire de Bézier.

Dans [76], l'algorithme de détection se base sur la vérification de deux conditions ; cette conjecture est formalisée puis démontrée dans [5]. La première condition consiste à trouver une direction privilégiée  $\mathbf{v}$  telle que le produit de toute normale à la surface avec  $\mathbf{v}$  soit positif et la seconde à vérifier que la projection parallèle à  $\mathbf{v}$  de la frontière de cette surface, vue comme courbe paramétrique, ne s'autointersecte pas.

Grinspun et Schröder [29] reprennent cette conjecture mais au lieu de trouver une direction privilégiée pour la première condition, ils montrent qu'il existe un plan de séparation entre l'origine et l'image de l'application de Gauss de la *patch* [15], c'est-à-dire que l'ensemble des normales à la surface est contenu dans un cône non-dégénéré (*i.e.*, un cône avec un demi-angle inférieur à  $\frac{\pi}{2}$ ). Si cette condition n'est pas vérifiée, la *patch* est subdivisée et la condition est vérifiée sur chaque sous-triangle. Le calcul du cône se base sur la diagonalisation de la matrice de subdivision donc sur les valeurs et vecteurs propres de cette matrice.

Dans [68], l'idée de cône de tangentes et de normales est introduite. Elle est

décrite dans [42]. Ces cônes sont calculés pour borner les directions des tangentes ou des normales de la surface. Dans la même idée, dans [43], le cône de visibilité est défini. C'est l'ensemble des vecteurs selon lesquels tout point de la surface peut être vu sans interférence.

Plusieurs méthodes de calcul de cône, circulaire [42, 68] ou pyramidale [12], ont été présentées dans la littérature ; toutes les surfaces considérées sont rectangulaires.

# Chapitre 3

## Description de la réalisation

### 3.1 Ensemble *QuasiNURBS*

Tel que mentionné dans la section 2.2, l’interpolation transfinie peut être utilisée d’une manière différente, pour fournir la définition d’un ensemble valide dans le contexte de l’étude de la robustesse. L’interpolant transfini, ne sera pas calculé explicitement mais est introduit uniquement dans le but de prouver à travers des théorèmes rigoureux, l’existence d’un unique ensemble valide, ou réalisation, qui sera une interprétation des données incohérentes fournies à une méthode numérique donnée.

Appelons  $S$  l’ensemble décrit par les entrées incohérentes  $\Delta$  définies à la section 1.1. Nous supposons ici que les données topologiques sont exactes et nous modifierons “légèrement” les données géométriques, que nous supposerons incertaines de manière à les faire concorder entre elles et avec les données topologiques. De toute façon, vu l’incertitude sur les données, l’utilisateur s’attend à ce que l’objet ne soit pas fabriqué exactement tel qu’il le veut mais avec des petites perturbations.

### 3.1.1 Définition et qualité de la réalisation : l'ensemble *QuasiNURBS*

Les frontières de  $S$  seront formées par une perturbation des sous-*patches NURBS* (ou *trimmed NURBS patches*) : elles seront cohérentes avec les sommets explicites géométriques  $\mathbf{v}_i, i = 1, 2$  et la version “légèrement” perturbée de la courbe frontière explicite  $\mathbf{b}(t)$  et conformes aux données topologiques.

Une sous-*patch* (ou *trimmed patch*) est représentée par la fonction  $\mathbf{F}$  définie sur un domaine  $D_0$  de  $R^2$  et à valeurs dans  $R^3$  et des  $p$ -courbes  $\mathbf{p}^k(t), k = 1, \dots, m$  définies pour  $t \in [0, 1]$  et à valeurs dans le domaine paramétrique  $u-v$  de  $R^2$  (voir FIG.2.1). L'ensemble des  $m$   $p$ -courbes définit un domaine  $D \subseteq D_0$  en spécifiant la frontière  $\partial D$  de  $D$ . La frontière  $\partial D$  est une courbe fermée dont les  $m$  segments sont les  $p$ -courbes. Chaque fonction  $\mathbf{p}^k$  est supposée injective, chaque segment intersecte son voisin uniquement aux extrémités et deux  $p$ -courbes non-adjacentes ne s'intersectent nulle part. Nous pouvons vérifier ces hypothèses, du moins pour les courbes de Bézier, en utilisant les méthodes décrites dans [2]. Nous supposons aussi que si une frontière de la *trimmed patch* est définie par un segment de droite confondu avec un côté du domaine  $D_0$  (dans notre étude  $D_0 = [0, 1]^2$ ), alors il existe une  $p$ -courbe qui coïncide avec ce segment de droite (donc avec un segment ou une partie d'un segment frontière du pavé  $[0, 1]^2$ ).

Il serait préférable que la distance entre la courbe explicite  $\mathbf{b}^k(t)$  et l'image par  $\mathbf{F}$  de la  $p$ -courbe  $\mathbf{p}^k(t)$ , c'est-à-dire  $\|\mathbf{b}^k(t) - \mathbf{F}(\mathbf{p}^k(t))\|, 0 \leq t \leq 1$ , soit petite. En d'autres termes, pour  $t_0$  fixé, on aimerait garantir que les points  $\mathbf{b}^k(t_0)$  et  $\mathbf{F}(\mathbf{p}^k(t_0))$  de  $R^3$  ne soient pas éloignés. Si  $\mathbf{b}$  est linéaire par morceau, la paramétrisation sera faisable. Sinon, pour interpoler les points  $\mathbf{v}_i$  ( $i = 1, 2$ ) en minimisant cette distance, nous pouvons introduire une reparamétrisation de  $\mathbf{b}^k(t)$  [59, p. 241]. Dans le cas général où  $\mathbf{b}$  est une courbe spline, ce problème est complexe et nous ne l'avons pas traité.

En fait, nous parlons de reparamétrisation mais la nouvelle courbe, notons-la  $\beta^k(t)$ , ne sera pas exactement la courbe décrite par  $\mathbf{b}^k(t)$  mais une version

“légèrement” perturbée de celle-ci. Étant donné les extrémités  $\mathbf{v}_1$  et  $\mathbf{v}_2$ , correspondant à  $t = 0$  et  $t = 1$  respectivement, nous définissons les courbes  $\beta^k(t)$  à partir des courbes  $\mathbf{b}^k(t)$ ,  $t \in [0, 1]$  tel que  $\beta^k(0) = \mathbf{v}_1$  et  $\beta^k(1) = \mathbf{v}_2$ , par une simple interpolation linéaire donnée par :

$$\beta^k(t) = \mathbf{b}^k(t) + (1 - t)[\mathbf{v}_1 - \mathbf{b}^k(0)] + t[\mathbf{v}_2 - \mathbf{b}^k(1)] \quad (3.1)$$

On peut en déduire facilement l’écart entre la fonction  $\mathbf{b}(t)$  et la fonction  $\beta^k(t)$  (pour définir, maintenant, ce qu’on entend par “légèrement” perturbée) :

$$\|\beta^k(t) - \mathbf{b}^k(t)\| \leq \max\{\|\mathbf{v}_1 - \mathbf{b}^k(0)\|, \|\mathbf{v}_2 - \mathbf{b}^k(1)\|\}, 0 \leq t \leq 1. \quad (3.2)$$

L’écart entre la courbe donnée  $\mathbf{b}$  et la courbe perturbée à partir de  $\mathbf{b}$ , c’est-à-dire  $\beta$  n’est jamais plus grand que l’écart déjà existant dans les données. Cependant, il peut arriver que  $\beta^k = \mathbf{b}^k$ , dans le cas par exemple où  $\mathbf{b}^k$  est une courbe linéaire par morceau ou une courbe NURBS dont les points de contrôle, initial et final, sont les points  $\mathbf{v}_1$  et  $\mathbf{v}_2$ .

Nous pouvons, maintenant, donner les conditions qui permettent d’établir une correspondance point par point entre chacune des courbes  $\mathbf{F}(\mathbf{p}^k(t))$  (frontières de la *trimmed patch* donnée) et les courbes  $\beta^k(t)$  (c’est-à-dire les courbes  $\mathbf{b}^k(t)$  éventuellement perturbées) par la proposition suivante ([4]) :

**Proposition 3.1.1.** *Si, pour une seule patch,*

1. *la fonction  $\mathbf{F}$  est injective ;*
2. *pour chaque  $k$  : la fonction  $\mathbf{p}^k$  est injective, les  $p$ -courbes adjacentes ne s’intersectent qu’en leurs extrémités, et deux  $p$ -courbes non-adjacentes ne s’intersectent nulle part ;*
3. *pour chaque  $k$*

$$\left\| \frac{\mathbf{b}^k(t) - \mathbf{b}^k(t')}{t - t'} \right\| > (\|\mathbf{v}_1 - \mathbf{b}^k(0)\| + \|\mathbf{v}_2 - \mathbf{b}^k(1)\|), \quad t < t' ;$$

*alors, pour chaque  $k$ , les applications  $\mathbf{F}(\mathbf{p}^k(t)) \rightarrow \beta^k(t)$ , de l’ensemble image  $Im(\mathbf{F}(\mathbf{p}^k(t)))$  vers l’ensemble image  $Im(\beta^k(t))$ , sont bien définies et bijectives.*

*Démonstration*

Les hypothèses 1 et 2 de la proposition impliquent que  $\mathbf{F} \circ \mathbf{p}^k$  est injective sur  $[0, 1]$ , alors, pour chaque  $k$ , les applications  $\mathbf{F}(\mathbf{p}^k(t)) \rightarrow \beta^k(t)$  sont bien définies. La condition 3 assure que pour chaque  $k$ ,  $\beta^k(t)$  est injective. En effet, supposons que pour  $t < t'$ ,  $\beta^k(t) = \beta^k(t')$ , alors :

$$\mathbf{b}^k(t) - \mathbf{b}^k(t') - (t - t')(\mathbf{v}_1 - \mathbf{b}^k(0)) + (t - t')(\mathbf{v}_2 - \mathbf{b}^k(1)) = 0,$$

i.e.,

$$\frac{\mathbf{b}^k(t) - \mathbf{b}^k(t')}{t - t'} = (\mathbf{v}_1 - \mathbf{b}^k(0)) - (\mathbf{v}_2 - \mathbf{b}^k(1)),$$

soit,

$$\left\| \frac{\mathbf{b}^k(t) - \mathbf{b}^k(t')}{t - t'} \right\| \leq (\|\mathbf{v}_1 - \mathbf{b}^k(0)\| + \|\mathbf{v}_2 - \mathbf{b}^k(1)\|), \quad t < t'.$$

D'où la contradiction avec la condition 3 et par suite  $t < t'$ ,  $\beta^k(t) \neq \beta^k(t')$ , c'est-à-dire  $\beta^k(t)$  est injective. Alors les applications  $\mathbf{F}(\mathbf{p}^k(t)) \rightarrow \beta^k(t)$ , de l'ensemble image  $\text{Im}(\mathbf{F}(\mathbf{p}^k(t)))$  vers l'ensemble image  $\text{Im}(\beta^k(t))$ , sont bijectives.  $\square$

Avec les hypothèses de la proposition précédente, nous définissons la perturbation appliquée à la frontière de la *trimmed patch* donnée par :

$$\epsilon : \partial D \rightarrow R^3$$

telle que :

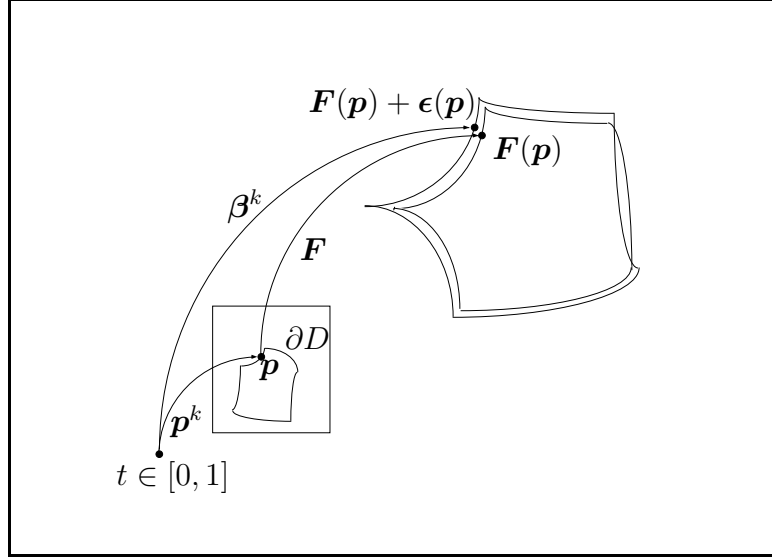
$$\epsilon(\mathbf{p}) = \beta^k([\mathbf{p}^k]^{-1}(\mathbf{p})) - \mathbf{F}(\mathbf{p}), \quad \mathbf{p} \in \partial D, \quad (3.3)$$

où  $[\mathbf{p}^k]^{-1}$  est l'inverse [66] de la  $p$ -courbe  $\mathbf{p}^k$  telle que, pour  $t$  donné dans  $[0, 1]$ ,  $\mathbf{p}^k(t) = \mathbf{p}$ .

Ainsi,  $\epsilon$  (fonction de  $\mathbf{p} \in \partial D$ ) est la différence entre la frontière de la *trimmed patch*  $\mathbf{F}[D]$ , et les courbes frontières (modifiées)  $\beta^k$  (voir FIG.3.1).

Ayant défini la fonction perturbation  $\epsilon$  sur la frontière de la *trimmed patch*, nous nous proposons maintenant de la prolonger sur tout le domaine  $D$  pour définir une nouvelle *trimmed patch*

$$\mathbf{F}(\mathbf{p}) + \epsilon(\mathbf{p}), \quad \mathbf{p} \in D,$$

FIG. 3.1: Définition de la perturbation sur  $\partial D$ 

qui interpole les courbes  $\beta$  en tout point de  $D$ . Ces nouveaux *trimmed patches* seront les frontières de l'ensemble *QuasiNURBS*  $S$ , *i.e.*, de la réalisation  $S$  dans  $R^3$  de la représentation donnée  $\Delta$ .

Notons  $\epsilon(\mathbf{p})$  une composante du vecteur  $\epsilon(\mathbf{p})$  et définissons la constante de Lipschitz par :

$$L = \sup_{\mathbf{p}_1, \mathbf{p}_2 \in \partial D, \mathbf{p}_1 \neq \mathbf{p}_2} \frac{|\epsilon(\mathbf{p}_1) - \epsilon(\mathbf{p}_2)|}{\|\mathbf{p}_1 - \mathbf{p}_2\|}. \quad (3.4)$$

On applique le théorème de Whitney [77] pour prolonger la fonction réelle  $\epsilon$  en une fonction continue sur tout  $R^2$  et le théorème de McShane pour prouver qu'elle préserve la condition de Lipschitz  $L$ .

Considérons les fonctions suivantes :

$$l(\mathbf{p}) = \sup_{\mathbf{q} \in \partial D} \{\epsilon(\mathbf{q}) - L \cdot \|\mathbf{p} - \mathbf{q}\|\}, \quad \mathbf{p} \in R^2, \quad (3.5)$$

$$u(\mathbf{p}) = \inf_{\mathbf{q} \in \partial D} \{\epsilon(\mathbf{q}) + L \cdot \|\mathbf{p} - \mathbf{q}\|\}, \quad \mathbf{p} \in R^2, \quad (3.6)$$

et

$$a(\mathbf{p}) = \frac{1}{2}[l(\mathbf{p}) + u(\mathbf{p})], \quad \mathbf{p} \in R^2. \quad (3.7)$$



On peut montrer d'une part [3], que toute fonction continue vérifiant la condition de Lipschitz sur  $R^2$  est encadrée inférieurement par la fonction  $l(\mathbf{p})$  et supérieurement par la fonction  $u(\mathbf{p})$  et d'autre part [52], que  $l(\mathbf{p})$  et  $u(\mathbf{p})$  sont elles-mêmes solutions au problème d'extension.

De plus,  $a(\mathbf{p})$  est une solution au problème d'extension, elle vérifie :

$$|a(\mathbf{p})| \leq \sup_{\mathbf{q} \in \partial D} |\epsilon(\mathbf{q})|, \quad \mathbf{p} \in R^2.$$

*Démonstration*

$$l(\mathbf{p}) = \sup_{\mathbf{q} \in \partial D} \{\epsilon(\mathbf{q}) - L \cdot \|\mathbf{p} - \mathbf{q}\|\} = l(\mathbf{p}) = \epsilon(\mathbf{q}_1) - L \cdot \|\mathbf{p} - \mathbf{q}_1\|,$$

où  $\mathbf{q}_1$  est le point de  $\partial D$  qui réalise la borne supérieure.

$$u(\mathbf{p}) = \inf_{\mathbf{q} \in \partial D} \{\epsilon(\mathbf{q}) + L \cdot \|\mathbf{p} - \mathbf{q}\|\} \leq \epsilon(\mathbf{q}_1) + L \cdot \|\mathbf{p} - \mathbf{q}_1\|$$

Donc,

$$\frac{1}{2}[l(\mathbf{p}) + u(\mathbf{p})] \leq \epsilon(\mathbf{q}_1) \leq |\epsilon(\mathbf{q}_1)| \leq \sup_{\mathbf{q} \in \partial D} |\epsilon(\mathbf{q})|.$$

De manière similaire, on obtient :

$$\frac{1}{2}[l(\mathbf{p}) + u(\mathbf{p})] \geq \epsilon(\mathbf{q}_2) \geq - \sup_{\mathbf{q} \in \partial D} |\epsilon(\mathbf{q})|,$$

où  $\mathbf{q}_2$  réalise la borne inférieure de  $\epsilon(\mathbf{q}) + L \cdot \|\mathbf{p} - \mathbf{q}\|$  sur  $\partial D$ . D'où,

$$|a(\mathbf{p})| \leq \sup_{\mathbf{q} \in \partial D} |\epsilon(\mathbf{q})|, \quad \mathbf{p} \in R^2.$$

□

Nous prolongerons donc  $\epsilon(\mathbf{p})$  à tout le domaine  $D$  en utilisant la fonction  $a(\mathbf{p})$  définie dans (3.7) :

$$\epsilon(\mathbf{p}) = \frac{1}{2}[l(\mathbf{p}) + u(\mathbf{p})], \quad \mathbf{p} \in D. \quad (3.8)$$

Ainsi, nous avons défini une face  $\mathbf{F}(\mathbf{p}) + \epsilon(\mathbf{p})$ ,  $\mathbf{p} \in D$ , de l'ensemble *QuasiNURBS*  $S$ . La différence entre cette face et la *trimmed patch* donnée  $\mathbf{F}[D]$  (compris dans la représentation donnée  $\Delta$ ), au sens de la métrique de Hausdorff, ne dépasse pas  $\sup_{\mathbf{q} \in \partial D} |\epsilon(\mathbf{q})|$ , *i.e.*, l'erreur à l'intérieur de la *patch* reste limitée par celle au bord. On peut résumer ces résultats dans une deuxième proposition [4].

**Proposition 3.1.2.** *La fonction  $\epsilon(\mathbf{p})$  donnée dans (3.8) est une fonction continue sur  $D$  qui est égale à la fonction (3.3) définie sur la frontière  $\partial D$ . De plus, la fonction définie dans (3.8) vérifie la condition de Lipschitz partout dans  $D$ , avec  $L$  (définie dans (3.4)) comme constante de Lipschitz, ainsi que l'inéquation*

$$|\epsilon(\mathbf{p})| \leq \sup_{\mathbf{q} \in \partial D} |\epsilon(\mathbf{q})|, \quad \mathbf{p} \in D.$$

Les constantes de Lipschitz des composantes de  $\epsilon$  peuvent être utilisées pour définir une constante de Lipschitz associée à la fonction vectorielle  $\epsilon$ . Nous avons donc prolongé  $\epsilon$  sur tout le domaine  $D$ , définissant ainsi une nouvelle *trimmed patch*

$$\mathbf{F}(\mathbf{p}) + \epsilon(\mathbf{p}), \quad \mathbf{p} \in D,$$

qui interpole les courbes  $\beta$  en tout point de  $D$  ainsi que les bornes de la perturbation  $\epsilon$ . Et si nous appliquons le même processus à toutes les *patches*, ces dernières s'ajusteront parfaitement au niveau des courbes frontières perturbées.

### 3.1.2 Représentation valide

On résume maintenant, dans une définition, les conditions introduites précédemment pour que la représentation soit **valide** ou **bien formée**.

**Définition 3.1.1.** *On suppose que les données symboliques ou topologiques sont exactes. On dit que la représentation  $\Delta$  est valide si :*

1. *les données topologiques définissent un 2-cycle topologique,*
2. *à chaque donnée topologique correspond une et une seule donnée géométrique dans  $R^3$  : à chaque 2-cell dans les données topologiques correspond une sous-patch de  $R^3$ , à chaque 1-cell correspond une courbe frontière  $\mathbf{b}(t)$  explicite entre les deux faces adjacentes et à chaque 0-cell dans les données symboliques correspond un point explicite géométrique  $\mathbf{v}$ . D'autre part, pour chaque trimmed patch, une et une seule  $p$ -courbe est associée à chaque 1-cell de la frontière du 2-cell correspondant à la trimmed patch,*
3. *les données géométriques vérifient les hypothèses de la proposition 3.1.1,*

4. la perturbation sur les données initiales n'introduit pas d'intersections incohérentes avec les données topologiques :

$$\mathbf{F}[D] + \epsilon[D] = \{\mathbf{F}(\mathbf{p}) + \epsilon(\mathbf{p}) : \mathbf{p} \in D\}$$

ne s'autointersecte pas, deux patches perturbées adjacentes

$$\mathbf{F}[D] + \epsilon[D] \text{ et } \mathbf{F}'[D'] + \epsilon'[D']$$

ne s'intersectent qu'en leur point ou courbe commune et deux patches perturbées non-adjacentes ne s'intersectent nulle part.

Avec ces conditions, la perturbation des données géométriques, décrite ci-haut, définit un plongement du 2-cycle topologique dans  $R^3$ . Pour résumer, un ensemble  $S$  est un plongement d'un 2-cycle topologique dans  $R^3$  s'il existe un homéomorphisme entre les données géométriques et les données topologiques. Intuitivement, cela voudrait dire que : chaque sommet dans les données topologiques correspond à un seul sommet dans les données géométriques, chaque arête à une seule courbe, chaque face à une seule *patch*. Si les *patches* sont représentées par des surfaces de Bézier, des critères pour obtenir la dernière condition de la définition 3.1.1 sont donnés dans [2].

Ainsi, si une représentation  $\Delta$  est valide, alors l'ensemble  $S$  (de  $R^3$ ) représenté par  $\Delta$  est bien défini.

**Définition 3.1.2.** *Si un ensemble  $S$  a une représentation  $\Delta$  valide, nous dirons que  $S$  est une réalisation de  $\Delta$  et nous noterons  $S \models \Delta$ .*

On pourra maintenant dire qu'un algorithme est robuste si, à partir d'une représentation donnée valide, il produit comme résultat une représentation valide et si ce résultat est le résultat d'un problème "très proche" du problème donné [35].

### 3.1.3 Exemple numérique

Illustrons maintenant le problème de robustesse sur les deux objets valides curvilinéaires que nous avons montré en exemple dans la figure du chapitre 1 (FIG. 1.1).

Considérons deux objets simples (FIG. 3.2). Ceci nous permet d'utiliser une version simple de l'algorithme d'intersection de Grandine-Klein [28]. Les résultats numériques ont été obtenus en utilisant la librairie DT\_NURBS [20] par Bisceglia<sup>1</sup>.

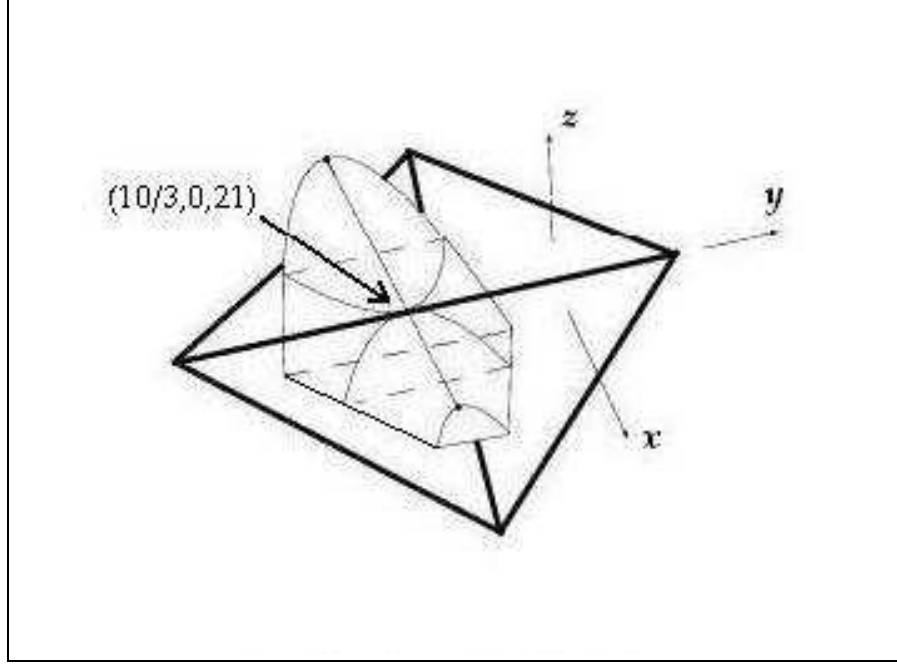


FIG. 3.2: Intersection de deux solides simples

Le premier solide, noté  $S_0$ , est un tétraèdre dont les faces sont triangulaires planes : une de ces faces est horizontale d'équation  $z = 21$  et une autre verticale dans le plan  $x = \frac{10}{3}$ , les deux autres faces de ce solide n'interviennent pas dans l'intersection. Les faces horizontales et verticales de cet objet sont donc définies de manière exacte par leurs sommets dont la 3<sup>ème</sup> composante est égale à 21 pour la première et la 1<sup>ère</sup> composante est la représentation flottante de  $\frac{10}{3}$ . Le second, noté  $S_1$ , est un objet ayant trois faces planes et une face courbe définie par une surface de Bézier. La face curvilinéaire est définie par :

$$\mathbf{F}(u, v) = \sum_{i=0}^1 \sum_{j=0}^3 \binom{1}{i} \binom{3}{j} u^i (1-u)^{1-i} v^j (1-v)^{3-j} \mathbf{P}_{ij}, \quad 0 \leq u, v \leq 1, \quad (3.9)$$

<sup>1</sup>J. Bisceglia a gracieusement accepté de faire pour nous ces calculs.

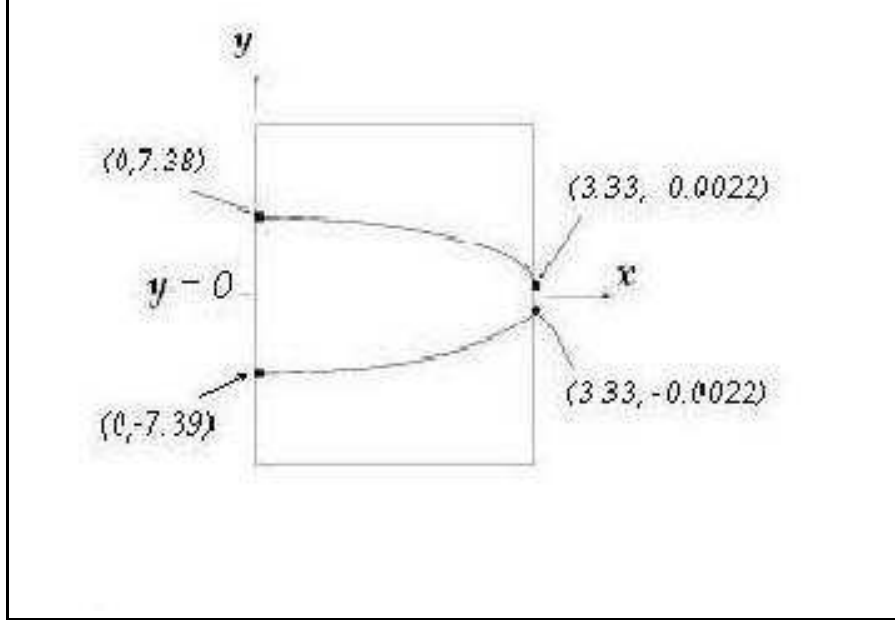


FIG. 3.3: Effet de l'arithmétique en virgule flottante : décalage du point d'intersection

où

$$\mathbf{P}_{00} = (0, -10, 0), \mathbf{P}_{01} = (0, -10, 40), \mathbf{P}_{02} = (0, 10, 40), \mathbf{P}_{03} = (0, 10, 0)$$

et

$$\mathbf{P}_{10} = (10, -1, 0), \mathbf{P}_{11} = (10, -1, 4), \mathbf{P}_{12} = (10, 1, 4), \mathbf{P}_{13} = (10, 1, 0).$$

Cette face est symétrique par rapport au plan  $y = 0$  et intersecte chacun des plans  $z = 21$  et  $x = \frac{10}{3}$  (voir FIG. 3.2). Il y a un seul point d'intersection, le point  $(10/3, 0, 21)$ , entre la face courbe et la droite donnée par

$$\lambda = \{P \in R^3 \mid P = (10/3, y, 21), y \in R\}.$$

Les autres faces de  $S_1$  n'intersectent pas la face courbe. La région intéressante pour nous se trouve au voisinage du point  $(10/3, 0, 21)$ .

L'algorithme de Grandine-Klein commence par déterminer la forme topologique des points  $(u, v)$  satisfaisant l'équation :

$$[0 \ 0 \ 1] \cdot \mathbf{F}(u, v) - 21 = 0 \tag{3.10}$$

qui se traduit par :

$$\sum_{j=1}^2 \binom{3}{j} v^j (1-v)^{3-j} [40(1-u) + 4u] - 21 = 0. \quad (3.11)$$

Cet algorithme calcule les  $p$ -courbes et les courbes correspondantes  $\mathbf{b}(t)$  pour l'intersection entre les deux surfaces.

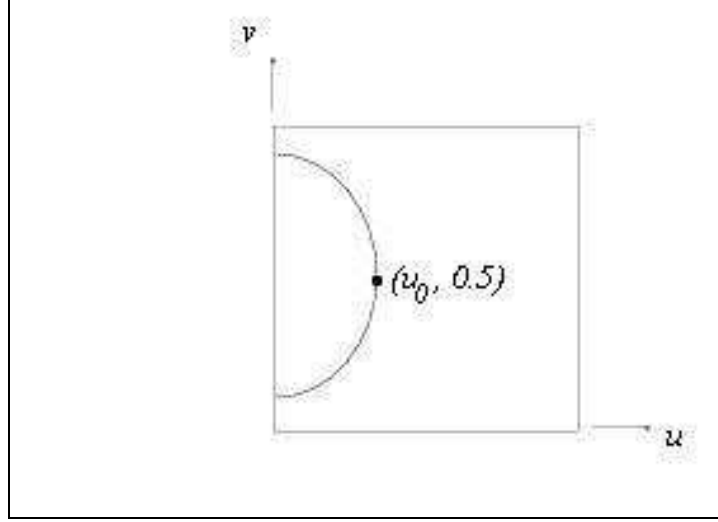


FIG. 3.4: Pré-image de l'intersection avec le plan  $z = 21$

Quand nous calculons l'intersection de  $\mathbf{F}$  avec la face triangulaire du plan  $z = 21$ , la courbe correspondante ne s'intersecte pas nécessairement avec le plan  $x = \frac{10}{3}$  une seule fois comme elle le devrait théoriquement mais une, deux ou trois fois. De même, le calcul de la courbe d'intersection de  $\mathbf{F}$  avec la face triangulaire du plan  $x = \frac{10}{3}$  ne donne pas un seul point mais un, deux ou trois points d'intersection avec le plan  $z = 21$ .

Ceci dépend largement de l'algorithme d'intersection de surface utilisé. Les routines D2PCUT et D2CLXT de DTNURBS [20, Sec. 4] ont été utilisées avec une précision de  $10^{-7}$ , pour trouver la courbe d'intersection  $\mathbf{b}(t)$ , B-spline d'ordre 4, appartenant exactement au plan  $z = 21$  (voir FIG. 3.3) et les deux parties de la  $p$ -courbe correspondante dans l'espace paramétrique  $u - v$ . En fait, la  $p$ -courbe est continue et atteint son maximum en un point qui dépasse le maximum exact

$(10/3, 0, 21)$  approximativement de  $3 \cdot 10^{-5}$  unités dans le sens positif de l'axe  $x$ . Cependant, comme on peut le voir (FIG. 3.3), l'intersection de  $\mathbf{b}(t)$  avec la droite  $x = \frac{10}{3}$  est assez loin du point exact d'intersection ( $0.0022$  unités) en  $y$ . Nous avons donc deux sommets au lieu d'un.

Il est possible de résoudre ce problème en perturbant la surface  $\mathbf{F}$  pour que l'intersection exacte coïncide avec l'intersection calculée représentée par deux  $p$ -courbes dans l'espace  $(u, v)$ . Il a été démontré récemment, au moins dans des cas particuliers [71], comment perturber les points de contrôle de deux surfaces données, de manière à faire coïncider l'intersection exacte de ces deux surfaces avec une solution calculée, représentée par deux courbes dans l'espace paramétrique relatif à chaque surface ou par une courbe explicite dans  $R^3$ . Cependant, dans le contexte des opérations sur les solides, la perturbation sur la surface doit tenir compte de l'intersection simultanée de plusieurs paires de surfaces et le problème général reste ouvert. Ici, nous devons tenir compte de trois surfaces en même temps. Cependant, dans cet exemple, la perturbation pour avoir l'intersection exacte entre les trois faces  $x = \frac{10}{3}$ ,  $z = 21$  et  $\mathbf{F}(u, v)$  est nettement simplifiée par la planarité de deux des faces de l'exemple qu'on a choisi.

Commençons par l'intersection entre la surface de Bézier (3.9) et le plan  $z = 21$ . Celle-ci est donnée par l'équation (3.11), qui se traduit par :

$$E = F^3(u, v) - 21 = 0. \quad (3.12)$$

Nous voulons trouver les nouveaux points de contrôle  $\mathbf{P}'_{ij}$  dont la troisième composante est une perturbation des points de contrôle originaux  $\mathbf{P}_{ij}$ , qui tiennent compte de la symétrie de  $\mathbf{F}$  par rapport à la droite  $y = 0$  et pour lesquels l'image des  $p$ -courbes est l'intersection *exacte* de la surface  $\mathbf{F}$  perturbée avec une version perturbée du plan  $z = 21$ , c'est à dire pour lesquels  $E = 0$ . Pour cela, nous remplacerons dans l'équation (3.12),  $u$  et  $v$  qui sont fonctions d'une variable  $\theta$ , ( $\theta \in [0, 1]$ ), par  $u(\theta)$  et  $v(\theta)$  respectivement. Dans le plan  $u-v$ , la courbe d'intersection peut être représentée par deux  $p$ -courbes obtenues en modifiant les résultats de D2PCUT. Nous avons choisi des quadratiques par morceau (FIG. 3.4). Elles

sont donc exprimées par :

$$p(\theta) = \begin{pmatrix} u(\theta) \\ v(\theta) \end{pmatrix} = \begin{Bmatrix} (1-\theta)^2 b_0^1 + 2\theta(1-\theta)b_1^1 + \theta^2 b_2^1 \\ (1-\theta)^2 b_0^2 + 2\theta(1-\theta)b_1^2 + \theta^2 b_2^2 \end{Bmatrix}$$

Nous remplaçons donc  $u$  et  $v$  par ces expressions dans l'équation (3.12) et nous obtenons un polynôme de degré 6 en  $\theta$ , soit :

$$E(\theta) = c_6 \theta^6 + \dots + c_1 \theta + c_0. \quad (3.13)$$

Les coefficients  $c_i$ ,  $i = 1, \dots, 6$ , sont fonctions des points de contrôle de la surface à perturber. Pour trouver les nouveaux points  $\mathbf{P}'_{ij}$ , nous devons avoir  $E = 0$  donc nous voulons que les coefficients  $c_i$ ,  $i = 1, \dots, 6$  soient nuls ou presque nuls, *i.e.*, pour  $\varepsilon$  petit,  $-\varepsilon \leq c_i \leq \varepsilon$ .

La perturbation des  $\mathbf{P}_{ij}$  obtenue est :

$$\begin{aligned} \mathbf{P}_{0j}^3 &\leftarrow \mathbf{P}_{0j}^3 + 0.824 \dots & j = 0, 3 \\ \mathbf{P}_{0j}^3 &\leftarrow \mathbf{P}_{0j}^3 - 0.747 \dots & j = 1, 2 \\ \mathbf{P}_{1j}^3 &\leftarrow \mathbf{P}_{1j}^3 + 0.863 \dots & j = 0, 3 \\ \mathbf{P}_{1j}^3 &\leftarrow \mathbf{P}_{1j}^3 + 0.654 \dots & j = 1, 2. \end{aligned}$$

Les coefficients  $c_i$ ,  $i = 1, \dots, 6$  sont majorés par  $\varepsilon = 0.00057$ .

De même, la pré-image de l'intersection entre  $\mathbf{F}$  et le plan  $x = \frac{10}{3}$  est le segment de droite  $u = \frac{1}{3}$  dans l'espace  $u - v$  et la première composante des  $\mathbf{P}_{ij}$  peut être perturbée (simultanément avec la perturbation de la troisième composante décrite plus haut) de manière telle que l'image de la  $p$ -courbe  $u = u_o$ ,  $u_o \approx \frac{1}{3}$  est l'intersection *exacte* entre la surface perturbée et une version perturbée du plan  $x = \frac{10}{3}$ . Dans ce cas, les perturbations sont données par :

$$\mathbf{P}_{0j}^1 \leftarrow \mathbf{P}_{0j}^1 + \frac{30u_o - 10}{3u_o - 3}, \quad \mathbf{P}_{1j}^1 \leftarrow \mathbf{P}_{1j}^1, \quad j = 0, 1, 2, 3$$

et le plan n'est pas perturbé.

Les perturbations simultanées, décrites ci-dessus, définissent deux ensembles  $S'_0$  et  $S'_1$  tels que  $S'_0$  est proche de  $S_0$  et  $S'_1$  est proche de  $S_1$  au sens de la métrique de Hausdorff et l'image des  $p$ -courbes dans l'espace paramétrique définit *exactement* l'intersection  $S'_0 \cap^* S'_1$ . Il faut noter, de plus, que la valeur de la constante de Lipschitz obtenue dans l'exemple numérique ( $L = 0.01114$ ), est prometteuse dans



le sens de garantir que les deux versions perturbées  $\mathbf{F} + \boldsymbol{\epsilon}$  et  $\mathbf{F}' + \boldsymbol{\epsilon}'$  s'ajustent sans intersections parasites. Pour obtenir de telles garanties, il est nécessaire de trouver l'application de Gauss associée à  $\mathbf{F} + \boldsymbol{\epsilon}$ , étant donnée l'application de Gauss associée à  $\mathbf{F}$ . L'analyse suivante apparaît dans [4].

Considérons par exemple le cas où  $\mathbf{F}$  est une face plane et pour simplifier, supposons que le domaine paramétrique soit le plan lui-même :

$$\mathbf{F}(u, v) = Id(u, v)$$

et que  $\mathbf{F}(\mathbf{p}(t))$  est une courbe plane simple fermée  $t \mapsto \boldsymbol{\alpha}(t)$ . Supposons que nous ayons aussi une courbe similaire dans  $R^3$ , perturbée par  $\boldsymbol{\epsilon}(t)$  :

$$t \mapsto \begin{bmatrix} \boldsymbol{\alpha}(t) \\ 0 \end{bmatrix} + \boldsymbol{\epsilon}(t) \in R^3$$

où on sait que  $\|\boldsymbol{\epsilon}(t)\| \leq \epsilon_{max}$ , et que pour chaque composante  $\epsilon$  de  $\boldsymbol{\epsilon}$ , on a :

$$|\epsilon(u_1, v_1) - \epsilon(u_2, v_2)| \leq L \cdot \|(u_1, v_1) - (u_2, v_2)\|$$

pour  $(u_1, v_1), (u_2, v_2)$  sur la courbe  $\mathbf{F}(\mathbf{p}(t))$ , comme dans la section 3.1.1.

Prolongeons l'application  $(u, v) \mapsto \boldsymbol{\epsilon}(u, v)$ , qui est définie sur la courbe  $t \mapsto \boldsymbol{\alpha}(t)$ , à la région limitée par la courbe  $t \mapsto \boldsymbol{\alpha}(t)$ , tel que nous l'avons décrit dans la section 3.1.1. Nous obtenons une application  $\mathbf{G} = \mathbf{F} + \boldsymbol{\epsilon}$  :

$$\mathbf{G}(u, v) = \begin{bmatrix} u \\ v \\ 0 \end{bmatrix} + \boldsymbol{\epsilon}(u, v).$$

Supposons qu'on ait l'information concernant les normales aux *patches* adjacentes à  $\mathbf{F}$ . Pour garantir que  $\mathbf{G}$  n'a pas d'intersections imprévues avec les *patches* adjacentes, on souhaiterait borner l'angle entre la normale unitaire  $\mathbf{n} = \mathbf{n}(u, v)$  de  $\mathbf{G}$  et la normale unitaire  $(0, 0, 1)$  de la surface plane  $\mathbf{F}$ . Sachant que :

$$\frac{\partial \mathbf{G}}{\partial u} = (1, 0, 0) + \frac{\partial \boldsymbol{\epsilon}}{\partial u} \quad \text{et} \quad \frac{\partial \mathbf{G}}{\partial v} = (0, 1, 0) + \frac{\partial \boldsymbol{\epsilon}}{\partial v},$$

nous calculons  $\frac{\partial \mathbf{G}}{\partial u} \times \frac{\partial \mathbf{G}}{\partial v}$  pour obtenir la normale à la surface  $\mathbf{G}$  puis nous la normalisons. On obtient un vecteur  $\mathbf{n}$  dont la troisième composante est supérieure

à  $1 - L^2 - O(L^3)$ , puisque  $\epsilon$  est différentiable par morceau et chaque composante de  $\frac{\partial \epsilon}{\partial u}$  et  $\frac{\partial \epsilon}{\partial v}$  est bornée par  $L$ .

Si on prend  $L = 0.01114$ , cela nous donne une borne inférieure du produit entre  $\mathbf{n}$  et  $(0, 0, 1)$  dont la valeur est 0.99987 et une borne supérieure de l'angle entre les deux vecteurs  $\sqrt{2}L \approx 0.01575$  *radians*.

L'exemple illustre que, pour avoir des bornes qui garantissent des *patches* adjacentes sans intersections parasites, il est important que la méthode numérique contrôle non seulement la “taille” de  $\epsilon$  mais aussi de sa dérivée, c'est à dire que la méthode doit contrôler la valeur de  $L$  aussi.

Le calcul des bornes sur l'angle entre la normale unitaire de  $\mathbf{G}$  et la normale unitaire de  $\mathbf{F}$  sera généralisé à la section 5.2.

## 3.2 Approche pour les surfaces subdivisées

Dans ce qui précède, nous avons montré comment on pouvait utiliser le théorème de Whitney pour construire une interpolation transfinie qui permet de valider la représentation des données incertaines dans le contexte des *trimmed NURBS*. Dans cette section, nous voulons montrer comment appliquer le théorème de Whitney dans le cas des *trimmed patches* de subdivision en présence d'incertitude sur les données.

Considérons la représentation permettant de découper les surfaces de subdivision, proposée par [50] et citée dans la section 2.3. Nous allons décrire les grandes lignes du processus de manière à mettre en évidence que, malgré que toutes les étapes soient bien accomplies, une incohérence dans les données existe et l'application du théorème d'extension de Whitney est justifiée. La représentation de [50] est basée sur un schéma de subdivision combinée [47]. Ce schéma permet la construction de surfaces de subdivision avec des courbes frontières arbitraires en tant que ces courbes sont  $C^1$ -continues par morceau, paramétrées et possèdent une procédure d'évaluation. De tels schémas modifient le procédé de subdivision (par exemple méthode de Loop ou Catmull-Clark) au voisinage d'une courbe pa-

ramétrique  $\mathbf{c} = \mathbf{c}(u)$  de *trimming*, fournie comme donnée, de manière à l’interpoler “exactement” [50].

Nous utiliserons l’expression subdivision combinée avec *trimming* (SCT) pour parler de la représentation proposée dans [50]. Pour faire une analogie entre cette dernière et la représentation *trimmed NURBS* décrite dans la section 3.1.1, on peut noter que la courbe frontière donnée  $\mathbf{c} : [0, 1] \rightarrow R^3$  correspond à la courbe fermée composée des  $m$  segments de courbes frontières  $\mathbf{b}^1(t), \dots, \mathbf{b}^m(t)$  reliés entre eux de manière à former une courbe fermée simple de  $R^3$ . La seule condition sur  $\mathbf{c}$  étant d’être  $C^1$ -continue par morceau, il est possible d’avoir des points anguleux comme les points de jonction entre deux courbes  $\mathbf{b}^{l-1}(1) = \mathbf{b}^l(0)$ , ( $l = 1, \dots, m \pmod{m}$ ) dans la représentation *trimmed NURBS*. Mais excepté cette similitude, les représentations sont tout à fait différentes. Une *patch* dans la représentation SCT est la limite du processus de subdivision combinée.

Ce processus permet de calculer une approximation de la surface initiale donnée et en l’absence d’erreurs d’arrondis, interpole exactement la courbe frontière  $\mathbf{c} = \mathbf{c}(u)$ . Notons qu’ici, l’interpolation transfinie se fait implicitement en appliquant le processus de subdivision pour calculer la *trimmed patch* en question alors que pour les *trimmed NURBS* décrites dans la section 3.1.1, l’interpolant transfini de Whitney est utilisé pour définir une face de la réalisation. Et nous voulons appliquer ce dernier au cas des représentations SCT.

Si on reprend [50], un point de contrôle quelconque en position  $i$  du maillage de la surface de subdivision au niveau  $j$  de la hiérarchie de subdivision est noté  $\mathbf{p}_i^j$ , et le paramètre associé dans le domaine de  $\mathbf{c}$  (*i.e.*  $[0, 1]$ ) est noté  $u_i^j$ . Une surface appelée surface originale, notons-la  $\Sigma$ , est représentée par les points de contrôle  $\mathbf{p}_i^0$  (donnés) [50] et une méthode de subdivision fixée en entrée telle que la méthode de subdivision de Loop [50, 47]. L’algorithme qui produit la *trimmed patch*, notons-la  $\Sigma_T$ , implique en premier lieu un remaillage local du polyèdre pour l’ajuster à la courbe donnée  $\mathbf{c}$  puis un échantillonnage de  $\Sigma$  pour choisir les points de contrôle de la *trimmed patch*. Et enfin, une étape d’approximation modifie la forme de la surface au voisinage de la courbe  $\mathbf{c}$  dans le but d’assurer un rapprochement vers

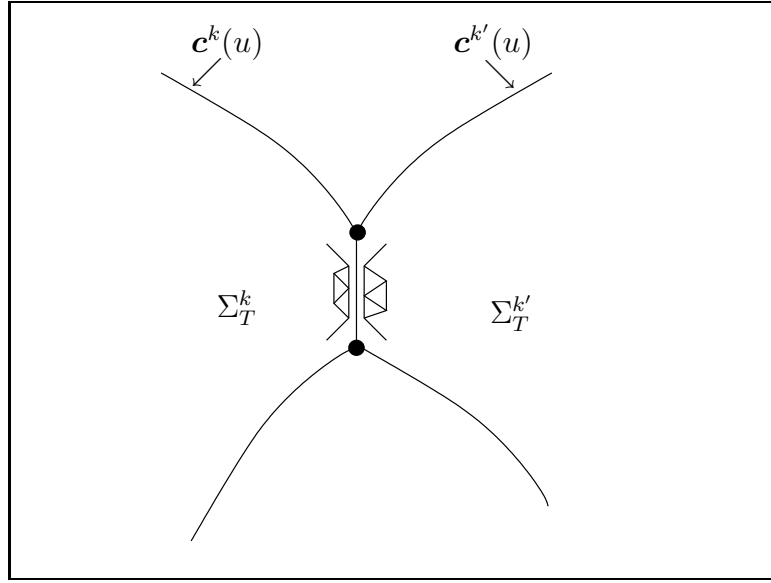
la surface  $\Sigma$ .

Le remaillage local modifiera normalement les paramètres initiaux  $u_i^0$  mais nous noterons encore ces paramètres  $u_i^0$  comme dans [50]. Puis la surface est “échantillonnée” pour trouver des points correspondant aux points de contrôle du maillage original.

La phase d’approximation dans l’algorithme de *trimming* ajuste la *trimmed patch*  $\Sigma_T$  à  $\Sigma$ . Cette opération n’est nécessaire qu’au voisinage de la courbe de *trimming*. En dehors de cette zone, la *trimmed patch*  $\Sigma_T$  est identique à  $\Sigma$ . Dans la région du bord, l’approximation de  $\Sigma$  est construite comme une hiérarchie de “coefficients de détail”  $\mathbf{d}_i^j$ , qui s’ajoutent aux points  $\mathbf{p}_i^j$  de contrôle pour apporter la modification nécessaire à l’ajustement. La profondeur de la hiérarchie de subdivision est limitée par un seuil de convergence [50, Sec. 3.3].

La *trimmed patch* finale est représentée alors par une hiérarchie de paramètres  $u_i^j$  qui dépendent des paramètres originaux et une hiérarchie de points de contrôle  $\mathbf{p}_i^j$  avec les vecteurs détails ajoutés  $\mathbf{d}_i^j$  qui dépendent des vecteurs originaux  $\mathbf{p}_i^0$ . Tel qu’observé dans [50], l’algorithme de *trimming* décrit ci-dessus, garantit une interpolation transfinie exacte de la *trimmed* courbe donnée. En fait, cette conclusion suppose que le processus de subdivision se poursuit jusqu’à la convergence et sans erreur d’arrondi. En pratique, cependant, ce ne sera pas le cas : la représentation sera obtenue après un nombre fini d’étapes du processus de subdivision et les calculs se feront en précision finie avec l’arithmétique en virgule flottante.

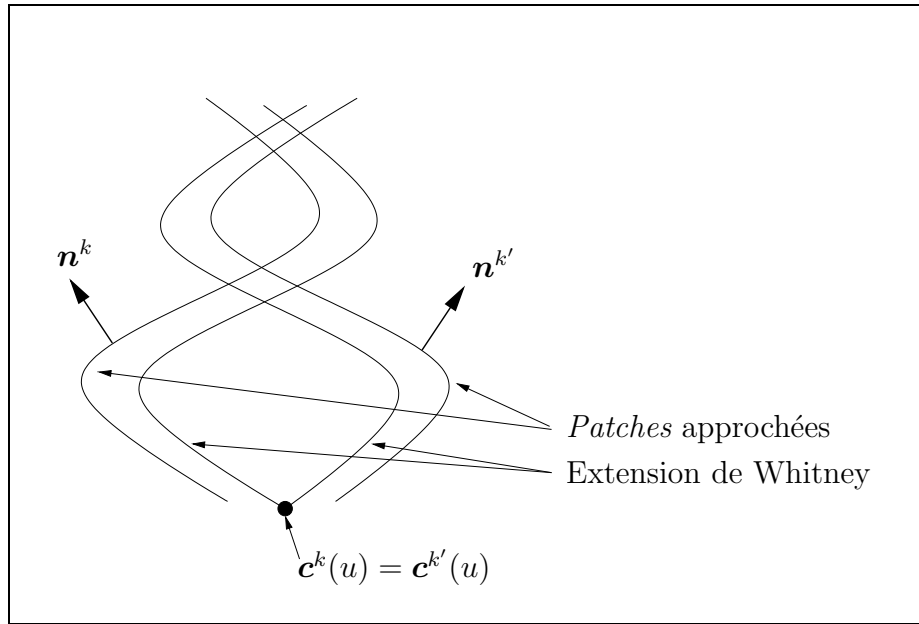
Il s’ensuit que, si on se met dans le contexte de la section 3.1 et si on veut prouver des théorèmes mathématiques rigoureux, énonçant que la représentation définit un sous-ensemble de  $R^3$  valide et dont les frontières sont, dans un certain sens, proche d’un ensemble de surfaces données  $\{\Sigma^k\}_{k=1,\dots,n}$ , alors il reste certains problèmes de définition. La situation est illustrée dans la figure suivante (FIG. 3.5), où les deux *patches* SCT sont censées se joindre le long d’une arête. En fait, on suppose que  $\mathbf{c}^k(u)$  et  $\mathbf{c}^{k'}(u)$  coïncident le long de l’arête entre  $\Sigma_T^k$  et  $\Sigma_T^{k'}$  mais en pratique, comme nous l’avons déjà mentionné, l’algorithme de *trimming* de  $\Sigma^k$  s’arrêtera après un nombre fini d’étapes et en plus, la courbe  $\mathbf{c}^k(u)$  sera évaluée

FIG. 3.5: Deux *trimmed patches* adjacentes obtenues par subdivision combinée

en précision finie ; par suite, la frontière de  $\Sigma_T^k$  ne coïncidera pas exactement avec  $c^k(u)$ .

D'autre part, il est possible de limiter la déviation de la *trimmed patch* par rapport à la surface originale, c'est-à-dire d'obtenir des bornes satisfaisantes (et ceci est implicite dans [50, p. 472]) de la différence, qu'on note  $\epsilon(\mathbf{p})$ , où  $\mathbf{p}$  est un point sur la frontière du polyèdre final de contrôle produit par le processus de subdivision combinée. En admettant que de telles bornes soient trouvées, le théorème d'extension de Whitney peut être appliqué exactement comme dans la section 3.1.1 pour donner une définition rigoureuse de l'ensemble concret de  $R^3$  représenté par les données incohérentes qui décrivent chacune des *trimmed patches*  $\Sigma_T^k$  et  $\Sigma_T^{k'}$  indépendamment. L'ensemble à partir duquel l'extension continue se fait, dans le théorème de Whitney, est ici la frontière du polyèdre final de contrôle.

Le seul problème qui reste, est la possibilité d'autointersections des *patches* SCT. En effet, le théorème de Whitney garantit une jonction correct des *patches* adjacentes en une arête commune mais n'exclut pas qu'il y ait des intersections indésirables entre eux (voir FIG. 3.6, où les *patches* SCT sont montrées de côté).

FIG. 3.6: Intersections indésirables de *patches* SCT adjacentes

Dans le cas des *patches* SCT basées sur la subdivision de Loop ou Catmull-Clark, il a été montré [72] que  $\Sigma_T^k$  (respectivement  $\Sigma_T^{k'}$ ) et ses dérivées peuvent être évaluées directement, ce qui veut dire que le vecteur normal  $n^k$  (respectivement  $n^{k'}$ ) peut être estimé. On peut alors appliquer la méthode de Volino-Thalmann [4, 76, 30] pour détecter une autointersection. C'est le sujet de la section 5 où une analyse sera présentée incluant les deux cas de *trimmed patches* (NURBS et subdivision combinée).

# Chapitre 4

## Vérification de non-autointersection

La vérification de non-autointersection est une des conditions pour avoir une représentation bien formée. Volino et Thalmann ont publié une conjecture intéressante [76] proposant de manière informelle des conditions suffisantes pour la non-autointersection de surfaces. Cette conjecture est utilisée dans différents domaines mais n'a pas été démontrée.

En fait, avant de la démontrer, nous voulons montrer, par des contre-exemples, que cette conjecture est incomplète et peut être mise à défaut dans certaines situations. Il est alors important de l'énoncer rigoureusement en précisant les hypothèses nécessaires pour obtenir la conclusion sur l'autointersection.

À cette fin, nous énoncerons dans la première section, la conjecture de Volino-Thalmann en précisant les points importants à investir pour un énoncé rigoureux, dans la seconde, nous donnerons nos contre-exemples et dans la troisième, nous énoncerons le critère d'autointersection sous forme de théorème permettant ainsi l'application du critère de Volino-Thalmann dans des situations plus générales.

Les différents points étudiés dans ce chapitre ainsi que la démonstration du théorème sont développés dans [5].

## 4.1 Formulation mathématique de la conjecture

La conjecture de Volino-Thalmann se base sur deux observations géométriques [76, p. C157]. Une autointersection peut avoir lieu si :

- La surface est suffisamment courbe pour former une boucle et toucher une autre partie de la surface (FIG. 4.1(a)).
- La projection du contour de la surface se superpose (FIG. 4.1(b)).

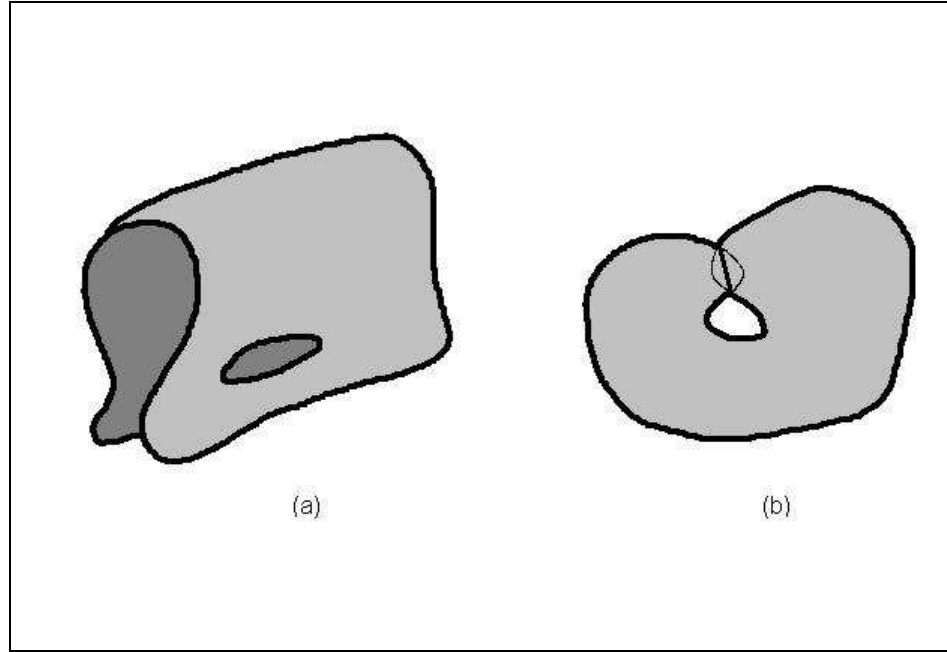


FIG. 4.1: Exemples d'autointersection [76]

Commençons par énoncer la conjecture telle qu'elle a été formulée dans [76].

### Conjecture 4.1.1. (Volino-Thalmann)

*Soient  $S$  une surface continue dans l'espace Euclidien, de frontière  $C$  et  $\mathbf{n}$  le vecteur normal à la surface en un point donné. S'il existe un vecteur  $\mathbf{v}$  tel que  $\mathbf{n} \cdot \mathbf{v} > 0$  "presque partout" sur  $S$ , et si la projection de  $C$  parallèlement à  $\mathbf{v}$  ne s'autointersecte pas, alors la surface  $S$  ne s'autointersecte pas.*



Intuitivement, la première condition suffit pour empêcher la première observation soit la FIG. 4.1(a) et la deuxième condition pour empêcher la seconde soit la FIG. 4.1(b).

La première chose à faire est de formaliser cet énoncé. Pour ce faire, il est primordial de préciser que la surface considérée est *paramétrique* :  $\mathbf{S} = \mathbf{S}(u, v)$ , où  $\mathbf{S}(u, v) \in R^3$  et  $(u, v) \in \overline{\Omega}$ ,  $\Omega$  étant un domaine paramétrique ouvert de  $R^2$ . Il y aura autointersection dans  $\mathbf{S}$  si et seulement si, pour deux points distincts  $\mathbf{p}_1$  et  $\mathbf{p}_2$  du domaine paramétrique  $\Omega$ , on a  $\mathbf{S}(\mathbf{p}_1) = \mathbf{S}(\mathbf{p}_2)$ , *i.e.*, si l'application  $\mathbf{S}$  n'est pas injective. On considère, sans perdre de généralité, que  $\mathbf{v}$  est vertical, *i.e.*,  $\mathbf{v} = [0, 0, 1]^T$ , où  $T$  indique la transposition. La condition  $\mathbf{n} \cdot \mathbf{v} > 0$  de la conjecture 4.1.1 se traduit alors par  $n_z > 0$  et la projection parallèle à  $\mathbf{v}$  devient une projection orthogonale dans le plan  $x$ - $y$ .

#### 4.1.1 Hypothèses générales

Dans la détection d'une autointersection, l'ordre de continuité de la surface considérée est une condition clé à préciser. Pour obtenir la conclusion de la conjecture 4.1.1, on doit introduire l'hypothèse que  $\mathbf{S}$  est  $C^1$ -continue, sauf peut-être en un nombre fini de points, au moins pour garantir l'existence du vecteur normal  $\mathbf{n} = \mathbf{n}(u, v)$ . D'autre part, la conjecture 4.1.1 est utilisée en pratique pour des surfaces composées en appliquant le test de détection d'autointersection à l'union des *patches* composant la surface [29]. Et dans le but de traiter ce cas, nous ferons seulement l'hypothèse que  $\mathbf{S}$  est  $C^1$ -continue par morceau. Par contre, il est important de choisir un domaine paramétrique commun à toutes les *trimmed patches* : la continuité de la surface dans l'espace Euclidien, comme énoncé dans la conjecture 4.1.1, *i.e.*, la continuité géométrique  $G^0$ , ne suffit plus. L'exemple 4.2.1 de la section 4.2 en est une preuve. Donc, quand nous disons "continuité", nous faisons référence à la continuité paramétrique.

Le domaine paramétrique  $\overline{\Omega}$  de la surface est  $\overline{\Omega} = \cup_{i=1}^n \overline{\Omega}_i$ , où les sous-domaines  $\Omega_i$ ,  $i = 1, \dots, n$  sont des ensembles ouverts, disjoints deux à deux, et simplement connexes. Un exemple de domaine non simplement connexe est discuté dans la

section 4.2 (exemple 4.2.4) et l'élaboration de l'hypothèse supplémentaire relative à ce cas ainsi que la preuve paraissent dans [5].

L'hypothèse de continuité “ $\mathbf{S}$  est  $C^1$ -continue par morceau” signifie en particulier que la frontière  $\partial\Omega_i$  de chaque sous-domaine  $\Omega_i$  est  $C^1$ -continue par morceau. On considère aussi que la fonction  $\mathbf{S}$  est continue sur tout son domaine et que les restrictions des dérivées au domaine  $\Omega_i$ ,  $i = 1, \dots, n$ , c'est-à-dire

$$\mathbf{S}_u|_{\Omega_i} \quad \text{et} \quad \mathbf{S}_v|_{\Omega_i} ,$$

sont continues et peuvent être prolongées en fonctions continues sur  $\overline{\Omega_i}$ , pour tout  $i$ .

**Remarque 4.1.1.** *Le prolongement de la dérivée à  $\overline{\Omega_i}$  ne coïncide pas nécessairement (et en général, ne coïncide pas du tout) avec le prolongement de la dérivée à  $\overline{\Omega_j}$  le long de la frontière entre les deux sous-domaines, soit  $\overline{\Omega_i} \cap \overline{\Omega_j}$ .*

Ces hypothèses étant posées, on introduit les notations :

$$\mathbf{S}(u, v) = \begin{bmatrix} f(u, v) \\ g(u, v) \\ h(u, v) \end{bmatrix} = \begin{bmatrix} \mathbf{F}(u, v) \\ h(u, v) \end{bmatrix} ,$$

où  $\mathbf{F} : \Omega \mapsto R^2$  est l'application donnée par  $\mathbf{F} = \begin{bmatrix} f \\ g \end{bmatrix}$ ,

le vecteur normal à  $\mathbf{S}$  au point  $(u, v)$

$$\mathbf{n}(u, v) = \mathbf{S}_u \times \mathbf{S}_v = \begin{pmatrix} g_u h_v - g_v h_u \\ h_u f_v - h_v f_u \\ f_u g_v - f_v g_u \end{pmatrix} = \begin{pmatrix} \frac{d(g, h)}{d(u, v)} \\ \frac{d(h, f)}{d(u, v)} \\ \frac{d(f, g)}{d(u, v)} \end{pmatrix} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} ,$$

et la matrice gradient de  $\mathbf{S}$  au point  $(u, v)$ .

$$A = ( \mathbf{S}_u \quad \mathbf{S}_v ) = \begin{pmatrix} f_u & f_v \\ g_u & g_v \\ h_u & h_v \end{pmatrix} = \begin{pmatrix} \nabla f \\ \nabla g \\ \nabla h \end{pmatrix} .$$

Nous ajouterons les précisions suivantes sur la matrice  $A$  et le vecteur normal  $\mathbf{n}$  car ces propriétés seront utilisées dans l'énoncé du théorème 4.3.1 de la section 4.3. La matrice  $A(u, v)$ , de dimension  $3 \times 2$ , est une approximation linéaire de l'application  $\mathbf{S}$  au point  $(u, v)$ . La matrice  $A$  est de rang maximal (rang=2)

si et seulement si  $\mathbf{n}(u, v) \neq \mathbf{0}$ . L'hypothèse  $\mathbf{n} \neq \mathbf{0}$  (ou  $A$  est de rang 2) est une condition naturelle pour exclure toute autointersection (voir par exemple [2]).

Selon les hypothèses déjà établies,  $\mathbf{n}|_{\Omega_i}$  et  $A|_{\Omega_i}$  sont continues et peuvent être prolongées continuellement à  $\overline{\Omega}_i$ ; on notera ces extensions  $\mathbf{n}_i$  et  $A_i$ , respectivement, pour tout  $i$ .

### 4.1.2 Cas d'un domaine simple

Si  $n = 1$ , le domaine paramétrique  $\Omega$  est composé d'une seule *patch*. On exprime la conjecture 4.1.1 de façon rigoureuse dans la proposition suivante.

**Proposition 4.1.1.** *Si*

1.  $\mathbf{S} : \overline{\Omega} \mapsto R^3$  est continue sur  $\overline{\Omega}$ ,
2.  $\mathbf{S}_u|_{\Omega}$  et  $\mathbf{S}_v|_{\Omega}$  sont continues et peuvent être prolongées en fonctions continues sur  $\overline{\Omega}$ ,
3.  $\mathbf{F}|_{\partial\Omega}$  est injective,
4.  $n_z > 0$  sur  $\Omega$ ,

alors l'application  $\mathbf{S}$  est injective.

La proposition 4.1.1 est similaire à la conjecture 4.1.1. L'hypothèse “ $\mathbf{F}|_{\partial\Omega}$  est injective” correspond à l'hypothèse “la projection de  $\mathbf{C}$  selon  $\mathbf{v}$  n'a pas d'autointersection” (rappelons que  $\mathbf{C} = \mathbf{S}[\partial\Omega]$ ). Cependant, il est important de noter que dans la proposition 4.1.1, on ne considère pas  $\mathbf{C}$  comme une simple courbe fermée dans le plan  $x$ - $y$  mais comme une fonction paramétrique qui doit être injective.

### 4.1.3 Cas d'un domaine composé

Si  $n \neq 1$ , le domaine paramétrique est composé de sous-domaines et la proposition 4.1.1 se généralise à la proposition suivante.

**Proposition 4.1.2.** *Si*

1.  $\mathbf{S} : \overline{\Omega} \mapsto R^3$  est  $C^1$ -continue par morceau sur  $\overline{\Omega} = \cup_{i=1}^n \overline{\Omega}_i$ ,

2. les restrictions des dérivées au sous-domaine  $\Omega_i$  soit  $\mathbf{S}_u|_{\Omega_i}$  et  $\mathbf{S}_v|_{\Omega_i}$ , sont continues et peuvent être prolongées en fonctions continues sur  $\overline{\Omega}_i$ , pour tout  $i$ ,

3.  $\mathbf{F}|_{\partial\Omega}$  est injective,

4.  $\mathbf{F}|_{\partial\Omega_i}$  est localement injective pour chaque  $i$ ,

5.  $n_z > 0$  sur  $\Omega_i$ , pour tout  $i$ ,

alors l'application  $\mathbf{S} : \overline{\Omega} \mapsto R^3$  est injective.

Si on compare la proposition 4.1.1 avec la proposition 4.1.2, on voit qu'une nouvelle hypothèse a été ajoutée, c'est-à-dire que  $\mathbf{F}|_{\partial\Omega_i}$  est localement injective. En plus de cette nouvelle hypothèse, cette proposition montre que l'application de la conjecture aux surfaces composées nécessite deux autres conditions : chaque frontière  $\partial\Omega_i$  est  $C^1$ -continue par morceau et les restrictions des dérivées au sous-domaine  $\Omega_i$  soit  $\mathbf{S}_u|_{\Omega_i}$  et  $\mathbf{S}_v|_{\Omega_i}$ , doivent être continues et prolongeables en fonctions continues sur  $\overline{\Omega}_i$ , pour tout  $i$ .

On verra dans la section 4.3 que la condition  $n_z > 0$  sur  $\Omega_i$  peut être relaxée de manière à obtenir un théorème plus fort que cette proposition.

## 4.2 Contre-exemples

Dans cette section, nous donnons des contre-exemples prouvant d'une part que les hypothèses de la conjecture 4.1.1 sont importantes mais tout à fait insuffisantes et justifiant d'autre part l'importance d'avoir les hypothèses des propositions 4.1.1 et 4.1.2 pour conclure la non-autointersection. On verra, dans la section qui suit, qu'on peut généraliser ces propositions en ajoutant certaines hypothèses plus faibles et qui ne sont pas satisfaites dans les contre-exemples.

Le premier exemple montre que la continuité  $G^0$  ne suffit pas. En effet, dans cet exemple, les *patches* considérées sont définies dans deux domaines paramétriques différents, la continuité de la surface composée n'est donc pas  $C^0$  mais elle est continue dans  $R^3$  donc la continuité  $G^0$  est vérifiée. Nous verrons dans le deuxième

exemple que, même si les *patches* considérées sont définies sur un seul domaine, le critère est faux si les conditions 3 et 4 de la proposition 4.1.2 ne sont pas satisfaites. Dans le troisième exemple, l'objectif en est beaucoup plus mathématique que pratique puisque c'est un cas un peu extrême, pas très grave en pratique comme par exemple dans le domaine de la modélisation du tissu. Il nous permettra, cependant, de mettre en évidence que si la condition 5 n'est pas vérifiée, il y a autointersection. Cet exemple illustrera aussi qu'on peut relaxer cette hypothèse à condition d'en ajouter d'autres comme nous le verrons dans la section 4.3, ce qui généralisera le critère comme il a été développé dans [5]. Quant au quatrième exemple, on montre que l'hypothèse "simplement connexe" doit être ajoutée à la proposition 4.1.2 car si le domaine paramétrique n'est pas simplement connexe alors le critère, tel qu'il a été énoncé, peut induire en erreur.

**Exemple 4.2.1.** Soient  $\mathbf{S}(u, v)$  et  $\mathbf{S}'(s, t)$  deux applications telles que, pour  $-1 \leq u \leq 1$ ,  $0 \leq v \leq 1$ ,

$$\mathbf{S}(u, v) = \begin{pmatrix} f_1(u, v) \\ g_1(u, v) \\ h_1(u, v) \end{pmatrix} = \begin{pmatrix} u \\ v \\ e^{u-1} - 1 - e^{-1} \end{pmatrix},$$

et pour  $0 \leq s \leq 1$ ,  $0 \leq t \leq 1$

$$\mathbf{S}'(s, t) = \begin{pmatrix} f_2(s, t) \\ g_2(s, t) \\ h_2(s, t) \end{pmatrix} = \begin{pmatrix} 1 - s \\ g_2(s, t) \\ -e^{-(1-s)} \end{pmatrix}.$$

avec  $g_2(s, t) = (1 - t)(0.25(1 - s) + 0.75) + t(-0.25(1 - s) + 0.25)$ .

Les normales respectives sont données par :

$$\mathbf{n}_1(u, v) = \mathbf{S}_u \times \mathbf{S}_v = \begin{pmatrix} 1 \\ 0 \\ e^{u-1} \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -e^{u-1} \\ 0 \\ 1 \end{pmatrix}$$

et

$$\mathbf{n}_2(s, t) = \mathbf{S}'_s \times \mathbf{S}'_t$$

soit

$$\mathbf{n}_2(s, t) = \begin{pmatrix} -1 \\ +0.5t - 0.25 \\ -e^{s-1} \end{pmatrix} \times \begin{pmatrix} 0 \\ +0.5s - 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -e^{s-1}(0.5s - 1) \\ 0 \\ -0.5s + 1 \end{pmatrix}.$$

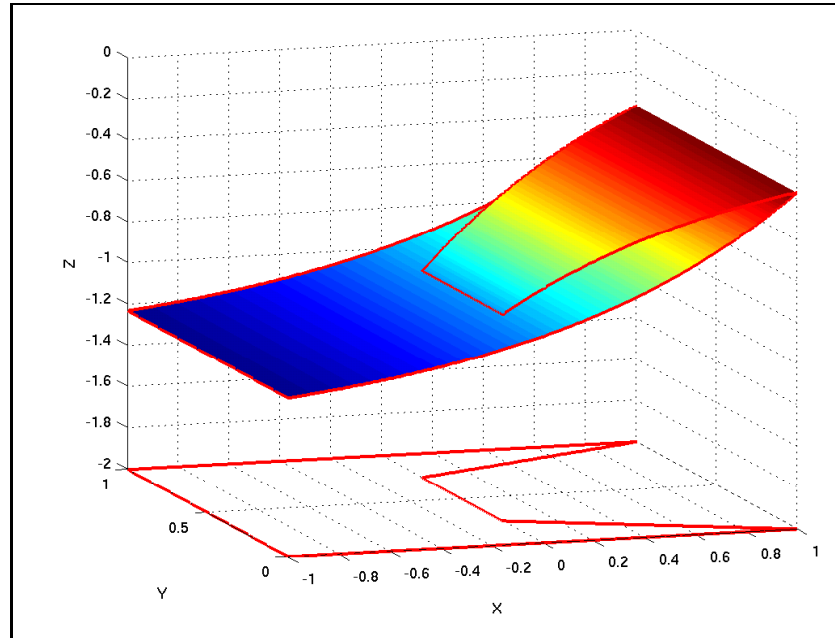


FIG. 4.2: Surface composée de deux *patches* qui s'autointersectent et sa projection

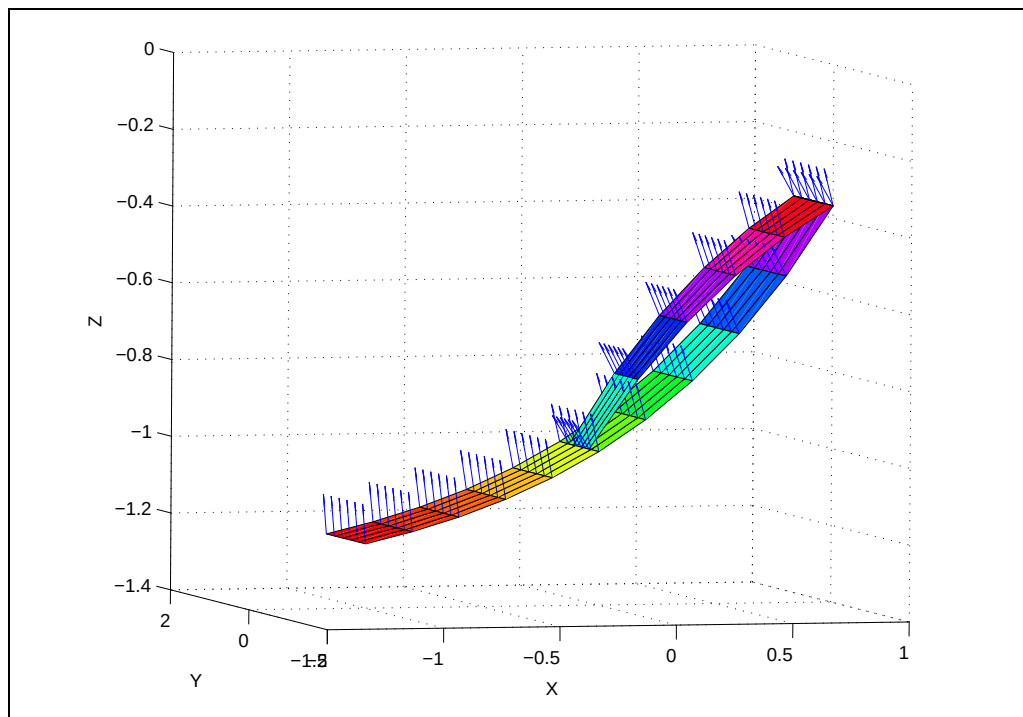


FIG. 4.3: Champ des normales de la surface composée de la figure 4.2

Pour chaque patch, la troisième composante de la normale est strictement positive donc les normales aux deux surfaces “pointent” vers le haut (voir FIG. 4.3). Les deux patches  $\mathbf{S}$  et  $\mathbf{S}'$  sont  $G^0$ -continues et la projection de la frontière ne s’autointersecte pas (FIG. 4.2). Les conditions du critère sont donc vérifiées et pourtant les deux patches s’intersectent en  $u = 0$  ( $s = 1$ ) et  $u = 1$  ( $s = 0$ ) (voir FIG. 4.2).

Cet exemple prouve que la condition 1 de la proposition 4.1.2 ne peut pas être négligée dans le cas des surfaces composées.

En fait, on pourrait donner un contre-exemple plus simple où la surface est composée de deux patches définies sur un même domaine et dont les paramétrisations sont simplement inversées. Ainsi, nous aurions pu à la place des deux surfaces courbes, prendre deux *patches* planaires qui se superposent bien que vérifiant les conditions de la conjecture 4.1.1. Mais les surfaces courbes ont été choisies pour plus de clarté dans l’illustration.

**Exemple 4.2.2.** Soit  $\mathbf{S}$  une surface définie sur le domaine  $D = [-1, 2] \times [0, 1]$  avec  $D = \cup_{i=1}^3 D_i$ , où  $D_1 = [-1, 0] \times [0, 1]$ ,  $D_2 = [0, 1] \times [0, 1]$  et  $D_3 = [1, 2] \times [0, 1]$  (Voir FIG. 4.4). Pour  $v \in [0, 1]$ , posons :

$$\begin{aligned} f(u, v) &= 0, \\ g(u, v) &= 1 - 2v, \\ h(u, v) &= \begin{cases} u, & -1 \leq u \leq 0 \\ 0, & 0 < u \leq 1 \\ u - 1, & 1 < u \leq 2 \end{cases} \end{aligned}.$$

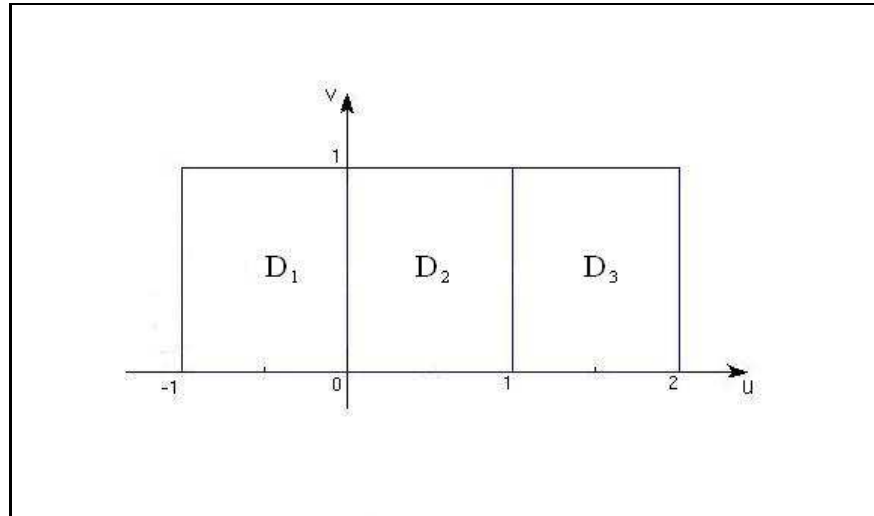
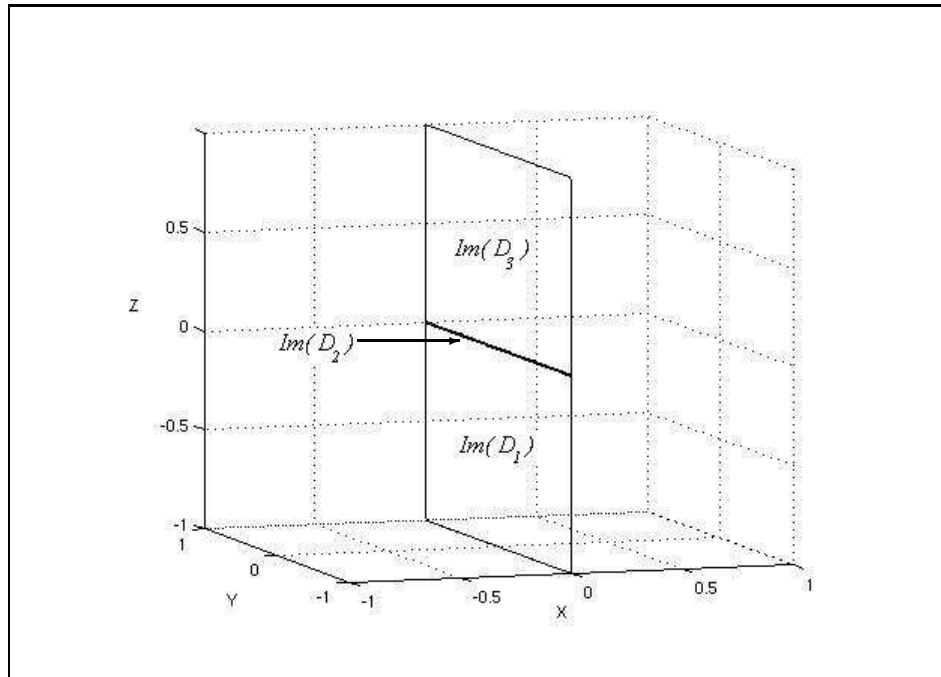
On a, pour  $-1 \leq u \leq 0$  et  $1 \leq u \leq 2$ ,

$$\frac{\partial \mathbf{S}}{\partial u} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

alors que pour  $0 \leq u \leq 1$ ,

$$\frac{\partial \mathbf{S}}{\partial u} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Donc,  $\frac{\partial \mathbf{S}}{\partial u}$  n’est pas continue sur tout  $D$  car non continue sur les frontières des sous-domaines. Ainsi,  $\mathbf{S}$  est  $C^1$ -continue à l’intérieur de chaque sous-patch, i.e.,

FIG. 4.4: Surface composée de trois sous-*patches* : domaine paramétriqueFIG. 4.5: Surface composée des trois *patches* dont l'une ( $Im(D_2)$ ) est dégénérée.



sur  $D_i$ ,  $i = 1, 2, 3$  mais seulement  $C^0$ -continue aux frontières des sous-patches, i.e., sur  $\partial D_i$ . Donc, les conditions 1 et 2 de la proposition 4.1.2 sur la continuité sont vérifiées. Et on a, pour tout  $u$  dans  $D$ ,

$$\frac{\partial \mathbf{S}}{\partial v} = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix}$$

Les normales à  $\mathbf{S}$  sur  $D_i$ ,  $i = 1, 2, 3$ , notons les  $\mathbf{n}_i$ , sont données par :

$$\mathbf{n}_i = \frac{\partial \mathbf{S}}{\partial u} \times \frac{\partial \mathbf{S}}{\partial v} \Big|_{D_i}.$$

Les normales à  $\mathbf{S}$  sur  $D_1$  et  $D_3$  sont égales et données pour  $i = 1, 3$  par :

$$\mathbf{n}_i = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}.$$

La normale à  $\mathbf{S}$  sur  $D_2$  n'existe pas. En effet,

$$\mathbf{n}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Ainsi, la condition de la conjecture sur les normales aux patches composant la surface, est vérifiée : les normales des deux patches sont de même sens et la projection de la frontière de la surface dans  $R^3$  ne s'autointersecte pas. Mais, malgré ces faits, il y a autointersection dans la surface composée : l'image de tout le sous-domaine  $D_2$  est un seul segment, le segment  $[-1, 1]$  sur l'axe  $y$  (voir FIG. 4.5) et l'image de chacun des deux côtés de la frontière de  $D_2$ , soit  $(u, 0)$ ,  $u \in [0, 1]$  et  $(u, 1)$ ,  $u \in [0, 1]$ , est réduite à un point, le point  $(0, -1, 0)$  et  $(0, 1, 0)$  respectivement. La condition 4 de la proposition 4.1.2 n'est donc pas respectée dans ce cas de figure ( $\mathbf{F}|_{\partial D_2}$  n'est pas localement injective) et la condition 3 ( $\mathbf{F}|_{\partial D}$  injective) non plus car deux segments de la frontière de  $D_2$  font partie de la frontière de  $D$ , et c'est ce qui a faussé le résultat de détection d'autointersection. Cet exemple montre qu'on ne peut pas éliminer les deux conditions 3 et 4.

L'exemple suivant montre que, même si les conditions 3 et 4 sont toutes les deux satisfaites, si la condition 5 sur la normale n'est pas satisfaite, il peut y avoir autointersection (à moins que d'autres conditions soient vérifiées comme nous le verrons dans la section 4.3).

**Exemple 4.2.3.** Soit  $\Omega_1 = \{(u, v) : -1 < u < 0, -1 < v < 1\}$  et

$$\Omega_2 = \{(u, v) : 0 < u < 1, -1 < v < 1\},$$

tels que

$$\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2 = [-1, 1] \times [-1, 1].$$

Et soit

$$f(u, v) = \begin{cases} f_1(u) + \left(v - \frac{1}{2}\right)^2 f_2(u), & \frac{1}{2} \leq v < 1 \\ f_1(u), & |v| < \frac{1}{2} \\ f_1(u) + \left(v + \frac{1}{2}\right)^2 f_2(u), & -1 < v \leq -\frac{1}{2}, \end{cases}$$

$$\text{où } f_1 \text{ est définie par } f_1(u) = \begin{cases} 0 & 0 \leq u < \frac{1}{2} \\ \left(u - \frac{1}{2}\right)^2 & \frac{1}{2} \leq u < 1 \end{cases},$$

avec  $f_1(-u) = -f_1(u)$  et  $f_2(u) = \frac{u^2}{2} \text{sgn}(u)$ .

De plus,  $g(u, v) = v$  et  $h(u, v) = |u|$ .

Alors, pour  $|u| \leq \frac{1}{2}$  et  $|v| \leq \frac{1}{2}$ , l'extension continue des vecteurs normaux pour  $u$  négatif et  $u$  positif respectivement sont les vecteurs :

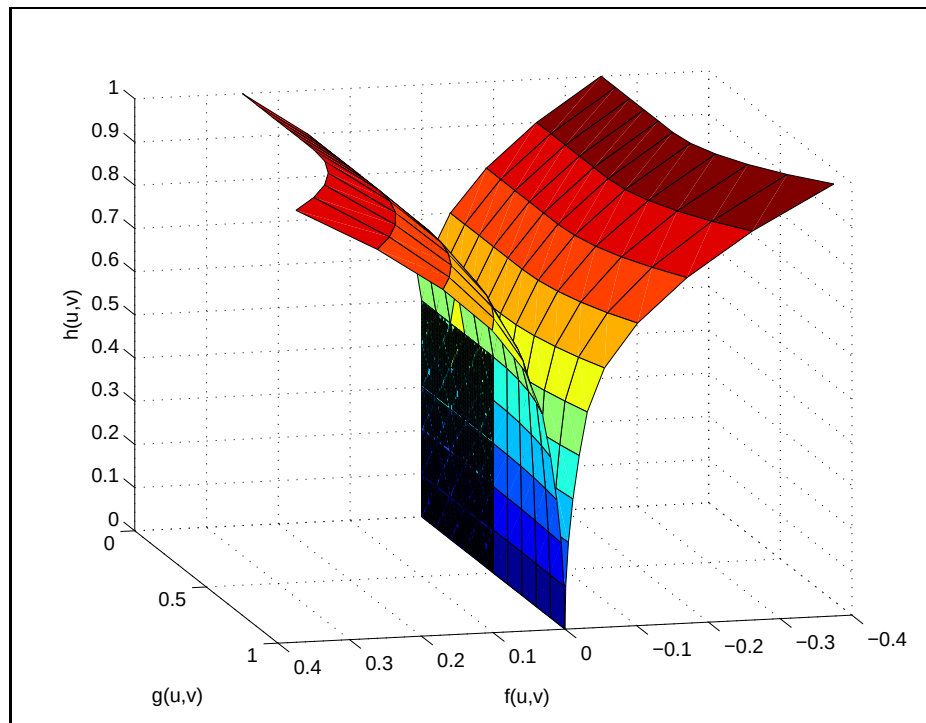
$$\mathbf{n}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{et} \quad \mathbf{n}_2 = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}.$$

Donc  $n_z = 0$ . Pour les autres valeurs de  $(u, v)$  dans  $[-1, 1] \times [-1, 1]$ ,  $\mathbf{n}_i$  est définie par

$$\mathbf{n}_i = \begin{pmatrix} -\sigma_i \\ \sigma_i(2v - \text{sgn}(v))f_2(u) \\ f'_1(u) + \left(v - \frac{\text{sgn}(v)}{2}\right)^2 f'_2(u) \end{pmatrix}$$

pour  $i = 1, 2$ , où  $\sigma_1 = -1$  et  $\sigma_2 = 1$ .

La surface  $\mathbf{S}$  est illustrée dans la figure suivante (FIG. 4.6), où seule la partie correspondant à  $v \geq 0$  est tracée, la partie correspondant à  $v \leq 0$  lui est symétrique. Les deux patches composant la surface se superposent dans la région définie

FIG. 4.6: Superposition de deux *patches* adjacentes

par  $|g(u, v)| < \frac{1}{2}$ ,  $0 < h(u, v) < \frac{1}{2}$ , qui correspond à la partie foncée sur la figure (FIG. 4.6), c'est-à-dire la région correspondant aux domaines

$$D_1 = \left\{ (u, v) : -\frac{1}{2} \leq u \leq 0, -\frac{1}{2} \leq v \leq \frac{1}{2} \right\}$$

et

$$D_2 = \left\{ (u, v) : 0 \leq u \leq \frac{1}{2}, -\frac{1}{2} \leq v \leq \frac{1}{2} \right\}.$$

Dans la région de la surface correspondant aux deux sous-domaines, la composante  $n_z$  de la normale est nulle et par suite, la surface est verticale dans ces deux régions. Mais comme ces deux sous-domaines sont adjacents, alors les deux parties verticales coïncident et il y a autointersection. En fait, les plans tangents sont dans cette région confondus. Il y a donc une hypothèse à ajouter dans ce sens à la proposition 4.1.2, comme on le verra dans la section 4.3.

**Exemple 4.2.4.** Soient dans l'espace paramétrique  $R^2$ , les domaines suivants,

$$D_1 = [-1, 1] \times [-1, 1], \quad \Gamma_1 = \partial D_1,$$

$$D_2 = \{(u, v) : -\frac{1}{2} + |v| < u < 0, |v| < \frac{1}{2}\}, \quad \Gamma_2 = -\partial D_2,$$

$$\Omega = D_1 \setminus \overline{D_2}, \quad \partial\Omega = \Gamma_1 + \Gamma_2.$$

Les courbes frontières sont orientées de manière telle que  $\Omega$  se trouve à gauche des vecteurs tangents (voir FIG. 4.7).

Et soit,

$$f(u, v) = \begin{cases} 4 - 1 + (u + 1)^{\frac{3-|v|}{\frac{1}{2}+|v|}}, & |v| < \frac{1}{2}, -1 < u < -\frac{1}{2} + |v| \\ u & |v| > \frac{1}{2} \text{ ou } (u > 0, -1 < v < 1) \end{cases}$$

$$g(u, v) = v$$

$$h(u, v) = \begin{cases} -(u + \frac{1}{2})(u + \frac{2}{3}), & -\frac{2}{3} < u < -\frac{1}{2} \\ -\frac{1}{4}|v|(|v| - \frac{1}{2})u(u + \frac{1}{2}), & -\frac{1}{2} < u < 0, |v| < \frac{1}{2} \\ 0 & \text{sinon.} \end{cases}$$

On peut vérifier que toutes les conditions de la conjecture 4.1.1 et aussi de la proposition 4.1.2 sont satisfaites.

Cependant, la surface s'autointersecte comme on peut l'observer sur la figure (FIG. 4.8). On a, par exemple,  $(u, v) = (-\frac{1}{2}, 0)$  et  $(u, v) = (\frac{1}{2}, 0)$  ont pour image

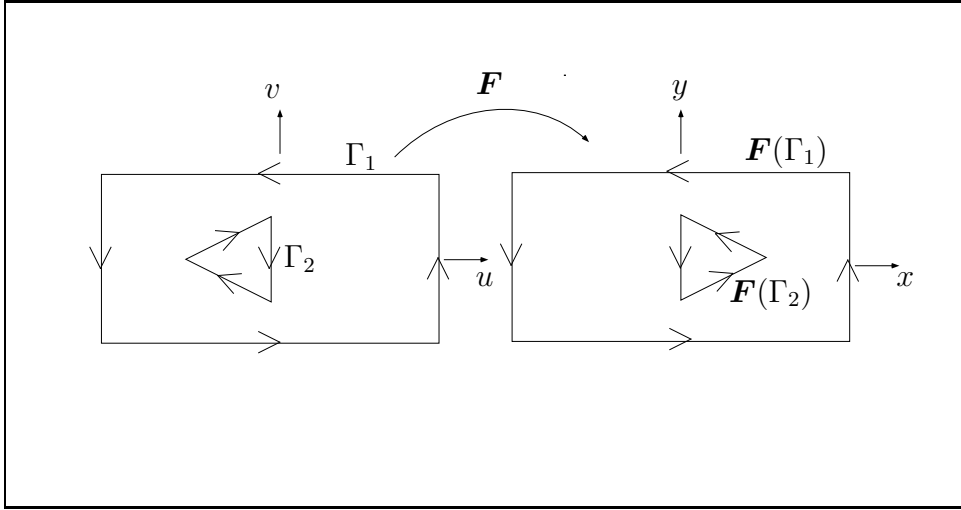


FIG. 4.7: Domaine paramétrique non simplement connexe

dans  $R^3$  le même point  $(\frac{1}{2}, 0, 0)^T$  et les deux points  $(u, v) = (-\frac{2}{3}, 0)$  et  $(u, v) = (0, 0)$  ont pour image  $(0, 0, 0)^T$ . En fait, on peut constater que l'image par  $\mathbf{F} : \overline{\Omega} \mapsto R^2$  de la courbe frontière  $\Gamma_1$  est la courbe  $\{(u, v, 0) : (u, v) \in \Gamma_1\}$  et que la courbe  $\Gamma_2$  a pour image la courbe  $\{(-u, v, 0) : (u, v) \in \Gamma_2\}$ , dont l'orientation est inversée, comme on peut le voir sur la figure précédente (FIG. 4.7). Intuitivement, on peut imaginer une surface d'eau, plane dont une partie (pour  $-\frac{2}{3} < u < -\frac{1}{2}$ ) émerge en vague pour retomber ensuite sur une autre partie de la surface plane. On note que les conditions de la conjecture 4.1.1 et aussi de la proposition 4.1.2 sont satisfaites à cause du fait qu'on a exclu du domaine l'image du trou  $D_2$  sinon la condition sur les normales ne sera pas vérifiée.

**Remarque 4.2.1.** La définition de  $h$  a été choisie de manière à rendre facilement visible l'autointersection. Le choix  $h(u, v) = 0$  pour tout  $(u, v)$  est largement suffisant pour donner un contre-exemple.

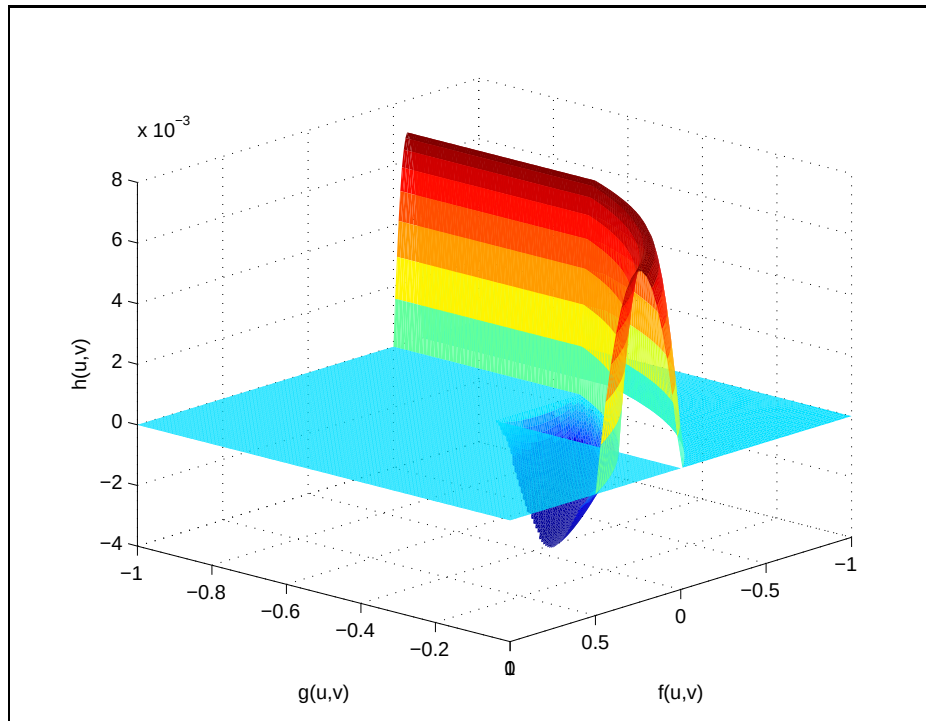


FIG. 4.8: Autointersection d'une surface dont le domaine est non simplement connexe

### 4.3 Énoncé du théorème

Dans la section 4.1.1, nous avons énoncé les hypothèses de base requises pour établir une condition suffisante de non-autointersection. L'échec du test de détection d'autointersection de Volino-Thalmann dans les contre-exemples de la section précédente, est justifié par la non-vérification de certaines hypothèses qui ont été ajoutées dans le théorème suivant [5]. Ce théorème généralise donc le critère de Volino-Thalmann dans le cas d'un domaine  $\Omega$  simplement connexe. Nous le réécrivons dans ce qui suit avec toutes ses hypothèses en reprenant les notations de la section 4.1.1.

**Théorème 4.3.1.** *Soit une surface  $\mathbf{S} : \overline{\Omega} \mapsto R^3$  continue sur  $\overline{\Omega}$ . Si les conditions suivantes sont vérifiées :*

- a.  $C^1$ -continue par morceau sur  $\overline{\Omega} = \cup_{i=1}^n \overline{\Omega}_i$ .
- b.  $\mathbf{S}_u|_{\Omega_i}$  et  $\mathbf{S}_v|_{\Omega_i}$ , sont continues et peuvent être prolongées en fonctions continues sur  $\overline{\Omega}_i$ , pour tout  $i$ .
- c.  $\mathbf{F}|_{\partial\Omega}$  est injective.
- d.  $\mathbf{n} \neq \mathbf{0}$ , ou encore, la matrice  $A$  est de rang maximal (rang=2).
- e.  $\mathbf{F}|_{\partial\Omega_i}$  est localement injective, pour tout  $i$ .
- f.  $n_z(u, v) = \frac{d(f,g)}{d(u,v)}(u, v) \geq 0$  où  $(u, v) \in \Omega_i$ , pour tout  $i$ .
- g. Pour chaque point  $\mathbf{p} \in \cup_{i=1}^n \partial\Omega_i$ , au moins une des conditions suivantes est vérifiée (voir FIG. 4.9) :
  - (1) Il existe un voisinage  $\mathcal{U}$  de  $\mathbf{p}$  tel que  $n_z > 0$  dans  $\mathcal{U} \setminus (\cup_{i=1}^n \partial\Omega_i)$ .
  - (2) L'application  $\mathcal{A} : R^2 \mapsto R^3$  définie par

$$\mathcal{A}(\mathbf{p}') = A_i(\mathbf{p}' - \mathbf{p}), \quad \mathbf{p}' \in D_i$$

où  $D_i \subset R^2$  est le secteur délimité par les vecteurs tangents à  $\partial\Omega_i$  au point  $\mathbf{p}$ , est injective.

alors, l'application  $\mathbf{S} : \overline{\Omega} \mapsto R^3$  est injective.

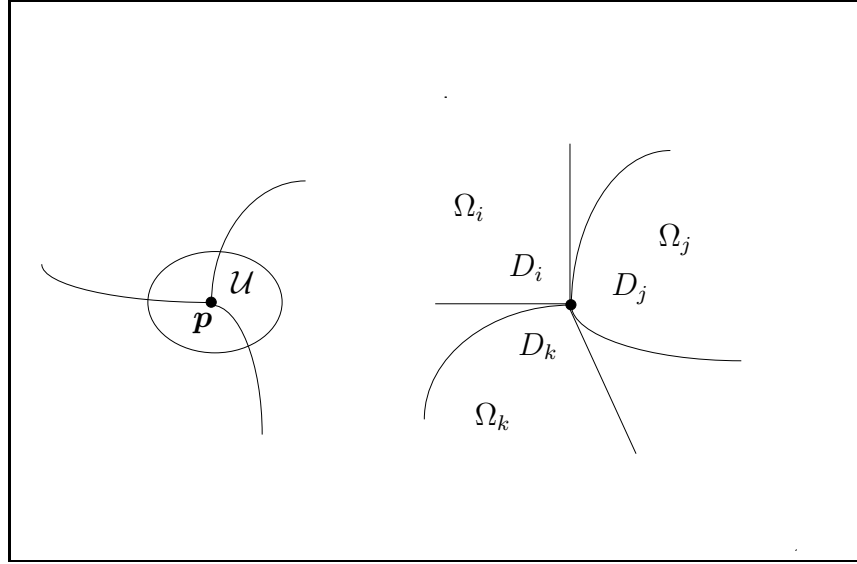


FIG. 4.9: Illustrations des conditions g1 et g2

La démonstration de ce théorème est donnée dans [5].

On notera que si  $n_z|_{\Omega_i} > 0$ , pour tout  $i$ , alors l'hypothèse g1 est vérifiée pour tout  $\mathbf{p} \in \cup_{i=1}^n \partial\Omega_i$ . Par conséquent, les hypothèses des deux propositions 4.1.1 et 4.1.2 sont suffisantes pour garantir que les conditions du théorème sont satisfaites.

Notons aussi que l'hypothèse g2 veut dire que l'approximation linéaire par morceau de  $\mathbf{S}$  au voisinage de  $\mathbf{p}$  est injective.

Les contre-exemples illustrent le fait que, malgré que les conditions du critère de Volino-Thalmann soient satisfaites, si certaines hypothèses du théorème ne sont pas vérifiées, il y aura autointersection. Pour chaque exemple, mettons en évidence la (ou les) hypothèse(s) du théorème non vérifiée(s) :

- Dans l'exemple 4.2.1, la condition de base sur  $\mathbf{S}$  (condition a) n'est pas vérifiée.
- Dans l'exemple 4.2.2, les hypothèses c et e ne sont pas satisfaites.
- Quant à l'exemple 4.2.3,  $\mathbf{n}_z = 0$  dans  $\Omega_i$  mais cette valeur pour  $\mathbf{n}_z$  est acceptable dans le théorème à condition qu'au moins une des deux conditions g1 ou g2 soit vérifiée, ce qui n'est pas le cas dans cet exemple. Et c'est la raison de la superposition des deux sous-*patches* adjacentes.



- Finalement, concernant l'exemple 4.2.4, toutes les hypothèses sur  $\mathbf{S}$  et  $\mathbf{F}$  sont satisfaites mais pas celle sur  $\Omega$ . En effet, le domaine n'est pas simplement connexe et le théorème ne s'applique plus.

Nous avons donc commenté ce théorème et illustré la suffisance de ses hypothèses et le fait qu'elles ne peuvent pas être éliminées. Mais la démonstration développée dans [5] le confirmera de façon rigoureuse avec les arguments mathématiques adéquats.

# Chapitre 5

## Non-intersection de *trimmed patches*

Nous avons vu au chapitre 3 (section 3.1.2) que pour montrer la validité d'une représentation *QuasiNURBS* d'un solide, les *trimmed patches* incluses dans cette représentation ne doivent pas s'autointersecter ni avoir des intersections indésirables avec les *trimmed patches* voisines.

Si les *patches* sont disjointes, la théorie pour la non-intersection des *trimmed patches* est bien établie bien que la mise en oeuvre ne soit pas nécessairement simple. Il suffirait, en effet, de comparer, dans ce cas, entre les ensembles englobants des *trimmed patches*.

Pour la vérification de la non-autointersection dans une *trimmed patch*, nous avons étudié le critère de Volino-Thalmann [76] (voir la théorie décrite au chapitre 4 et dans [5]).

Mais dans le cas où les deux *patches* sont adjacentes, Grinspun et Schröder énoncent qu'il suffirait d'appliquer le critère de Volino-Thalmann mais son application n'est pas du tout triviale si les *patches* adjacentes sont des *trimmed patches*. En effet, la paramétrisation de chaque *trimmed patch* sera alors définie sur un domaine paramétrique différent de l'autre (nous avons vu au chapitre 4 que cela ne suffit pas) et en plus du problème des domaines paramétriques indépendants, il y a le problème du *trimming*. Il faudra alors utiliser d'autres critères [42, 43] ou une

nouvelle méthode à développer.

## 5.1 Comment détecter une autointersection sur une surface perturbée

Les vecteurs normaux à une surface sont des indicateurs de la géométrie de la surface considérée, d'où leur importance en modélisation géométrique et en infographie.

En particulier, les méthodes de détection d'autointersection dans une *patch* et d'intersection entre *patches* dépendent du fait que nous connaissons des bornes sur les vecteurs normaux de la (ou des) *patch(es)* (voir Chap. 4 et [5, 42, 43, 64, 76]) ou sur l'application de Gauss associée à chaque *patch* [29]. Nous nous proposons donc de trouver une solution à cette partie du problème car dans tous les cas, c'est une partie centrale importante de la détection d'autointersection mais le problème global et général de détection reste ouvert comme on le verra dans la section 5.3.

Considérons donc les normales à une surface donnée. Il est possible de calculer les bornes sur la direction des normales à une surface donnée  $\mathbf{F}$  quand celle-ci est polynomiale [68] ou rationnelle [64]. Mais ce qui nous importe ici, ce sont les vecteurs normaux du quasi-interpolant  $\mathbf{G} = \mathbf{F} + \epsilon$ , *i.e.*, d'une face de l'ensemble *QuasiNURBS* que nous avons défini dans le chapitre 3.1.

la validité de la représentation, c'est-à-dire l'existence de la face de l'ensemble *QuasiNURBS*, dépend du respect de la géométrie de la surface originale  $\mathbf{F}$  décrite par les données topologiques. En d'autres termes, si, théoriquement, la *patch*  $\mathbf{F}$  ne comprend pas d'autointersection, la face correspondante de l'ensemble *QuasiNURBS*, obtenue après perturbation de  $\mathbf{F}$ , ne doit pas avoir d'autointersection non plus. Et si cette *patch* n'a pas d'intersections avec la *patch* adjacente autres que la courbe supposée commune (d'après les données topologiques), il devra en être de même pour les faces adjacentes correspondantes du *QuasiNURBS*. D'autre part, comme nous avons choisi d'utiliser le critère de Volino-Thalmann [76], l'étude sera concentrée sur les variations des normales de  $\mathbf{G}$ .

La question pour laquelle nous nous proposons de trouver une solution dans la section suivante, est donc :

Comment trouver une borne sur les vecteurs normaux de la *patch* perturbée  $\mathbf{G} = \mathbf{F} + \boldsymbol{\epsilon}$ , étant donné les vecteurs normaux de  $\mathbf{F}$ , *i.e.*, comment borner l'angle entre la normale à  $\mathbf{G}$ , notée  $\mathbf{n}_G$ , et la normale à  $\mathbf{F}$ , notée  $\mathbf{n}_F$ ?

Le résultat de l'étude qui suit, est présentée dans [74].

## 5.2 Bornes sur les normales de la *patch* perturbée

La première étape de détection d'autointersection est donc de calculer les bornes sur les variations de la normale à travers la surface perturbée  $\mathbf{G}$  par rapport à la surface originale  $\mathbf{F}$ , *i.e.*, en calculant l'angle, notons le  $\varphi$ , compris entre les deux normales associées  $\mathbf{n}_G$  et  $\mathbf{n}_F$ . À cet effet, notons  $\theta$  l'angle entre les vecteurs  $\mathbf{F}_u$  et  $\mathbf{F}_v$  et introduisons les notations traditionnelles suivantes :

$$\|\mathbf{F}_u\|^2 = \mathbf{F}_u \cdot \mathbf{F}_u = E \quad (5.1)$$

$$\|\mathbf{F}_v\|^2 = \mathbf{F}_v \cdot \mathbf{F}_v = G \quad (5.2)$$

$$\cos \theta \cdot \|\mathbf{F}_u\| \cdot \|\mathbf{F}_v\| = \mathbf{F}_u \cdot \mathbf{F}_v = F. \quad (5.3)$$

$E, F$  et  $G$  sont les coefficients de la première forme fondamentale de  $\mathbf{F} = \mathbf{F}(u, v)$  [49, 60]. On peut alors énoncer le théorème suivant, qui est une généralisation du résultat de [5] :

**Théorème 5.2.1.** *Soient  $\mathbf{F}$  une paramétrisation régulière donnée, représentant une face d'un objet et  $\mathbf{G} = \mathbf{F} + \boldsymbol{\epsilon}$  la face de l'ensemble QuasiNURBS associée où  $\boldsymbol{\epsilon}$  est la perturbation de Whitney appliquée à  $\mathbf{F}$ . On suppose que chaque composante de  $\boldsymbol{\epsilon}$  vérifie la condition de Lipschitz, *i.e.*,*

$$\left| \frac{\partial \epsilon_i}{\partial u} \right| \leq L \quad \text{et} \quad \left| \frac{\partial \epsilon_i}{\partial v} \right| \leq L, \quad i = 1, 2, 3.$$

*Alors, l'angle  $\varphi$  entre les normales  $\mathbf{n}_F$  et  $\mathbf{n}_G$  vérifie l'inégalité :*

$$\cos \varphi \geq 1 - \frac{L^2}{2} \left[ \frac{E + 2F + G}{EG - F^2} \right] \quad (5.4)$$

*Démonstration*

Considérons le quasi-interpolant  $\mathbf{G} = \mathbf{F} + \epsilon$  et  $\mathbf{F}$  la surface d'origine et notons  $\mathbf{n}_G$  et  $\mathbf{n}_F$  les normales à  $\mathbf{G}$  et  $\mathbf{F}$  respectivement, au point  $(u_0 + \delta u, v_0 + \delta v)$  et  $\mathbf{n}$  la normale à  $\mathbf{F}$  au point  $(u_0, v_0)$ .

Quand  $\delta u, \delta v \rightarrow 0$ , la projection de  $\mathbf{F}(u_0 + \delta u, v_0 + \delta v)$  sur le plan tangent à  $\mathbf{F}$  au point  $(u_0, v_0)$  est égale à  $\mathbf{F}(u_0, v_0)$  et la normale  $\mathbf{n}_F(u_0 + \delta u, v_0 + \delta v)$  est égale à  $\mathbf{n}(u_0, v_0)$ .

Posons  $\mathbf{f} = \mathbf{F}_u(u_0, v_0)$  et  $\mathbf{s} = \mathbf{n}(u_0, v_0) \times \mathbf{F}_u(u_0, v_0)$ . Dans le repère ortho-normé  $\left(\frac{\mathbf{f}}{\|\mathbf{f}\|}, \frac{\mathbf{s}}{\|\mathbf{s}\|}\right)$  du plan tangent à  $\mathbf{F}$  au point  $(u_0, v_0)$ , les coordonnées de  $\mathbf{F}(u_0 + \delta u, v_0 + \delta v)$  sont :

$$u = \mathbf{F}(u_0 + \delta u, v_0 + \delta v) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|}, \quad (5.5)$$

$$v = \mathbf{F}(u_0 + \delta u, v_0 + \delta v) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|}. \quad (5.6)$$

Considérons  $(u, v)$  comme nouveaux paramètres. Nous voulons définir l'application

$$\mathbf{m}(u, v) = (u_0 + \delta u, v_0 + \delta v)^T.$$

D'après la formule de Taylor, on a :

$$\mathbf{F}(u_0 + \delta u, v_0 + \delta v) \approx \mathbf{F}(u_0, v_0) + \delta u \cdot \underbrace{\mathbf{F}_u(u_0, v_0)}_{\mathbf{f}} + \delta v \cdot \mathbf{F}_v(u_0, v_0). \quad (5.7)$$

Alors, les équations (5.5) et (5.6) s'écrivent<sup>1</sup> :

$$\begin{aligned} u &= \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|} + \delta u \cdot \mathbf{f} \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|} + \delta v \cdot \mathbf{F}_v \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|}, \\ v &= \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} + \delta u \cdot \mathbf{f} \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} + \delta v \cdot \mathbf{F}_v \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|}. \end{aligned}$$

Comme :

$$\mathbf{F}_v \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|} = \cos \theta \cdot \|\mathbf{F}_v\|$$

où  $\theta = \angle(\mathbf{f}, \mathbf{F}_v)$  et

$$\mathbf{s} \cdot \mathbf{f} = 0; \quad \mathbf{F}_v \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} = \cos(\theta - \frac{\pi}{2}) \cdot \|\mathbf{F}_v\| = \sin \theta \cdot \|\mathbf{F}_v\|,$$

---

<sup>1</sup>Pour alléger le texte, on utilisera le signe  $=$  au lieu du signe  $\approx$  sauf pour un rappel. Cet abus de notation ne nuira pas au raisonnement puisque  $\delta u, \delta v \rightarrow 0$ .

alors,

$$\begin{aligned} u &= \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|} + \delta u \cdot \|\mathbf{f}\| + \delta v \cdot \cos \theta \cdot \|\mathbf{F}_v\|, \\ v &= \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} + \delta v \cdot \sin \theta \cdot \|\mathbf{F}_v\|. \end{aligned}$$

Soit, sous forme matricielle,

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \|\mathbf{F}_u\| & \cos \theta \cdot \|\mathbf{F}_v\| \\ 0 & \sin \theta \cdot \|\mathbf{F}_v\| \end{pmatrix} \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} + \begin{pmatrix} \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|} \\ \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} \end{pmatrix},$$

d'où on a :

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \begin{pmatrix} \|\mathbf{F}_u\| & \cos \theta \cdot \|\mathbf{F}_v\| \\ 0 & \sin \theta \cdot \|\mathbf{F}_v\| \end{pmatrix}^{-1} \begin{pmatrix} u - \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|} \\ v - \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} \end{pmatrix}.$$

Comme

$$\mathbf{m}(u, v) = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} u_0 + \delta u \\ v_0 + \delta v \end{pmatrix}$$

et  $\mathbf{F}$  est régulière (donc,  $\mathbf{F}_u$ ,  $\mathbf{F}_v$  et  $\sin \theta$  sont non nulles), on peut définir  $\mathbf{m}(u, v)$  par

$$\mathbf{m}(u, v) = \begin{pmatrix} \frac{1}{\|\mathbf{F}_u\|} & \frac{-\cos \theta}{\sin \theta \cdot \|\mathbf{F}_v\|} \\ 0 & \frac{1}{\sin \theta \cdot \|\mathbf{F}_v\|} \end{pmatrix} \begin{pmatrix} u - \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|} \\ v - \mathbf{F}(u_0, v_0) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix}. \quad (5.8)$$

Dans le repère orthonormé  $\left(\frac{\mathbf{f}}{\|\mathbf{f}\|}, \frac{\mathbf{s}}{\|\mathbf{s}\|}, \frac{\mathbf{n}}{\|\mathbf{n}\|}\right)$ , on approche  $\mathbf{F}(u_0 + \delta u, v_0 + \delta v)$  par sa projection sur le plan tangent soit le vecteur  $(u, v, 0)^T$  et le vecteur perturbation est donné par :

$$\epsilon(u_0 + \delta u, v_0 + \delta v) = \left( \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|}, \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|}, \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} \right)^T.$$

Donc, dans ce repère, le vecteur  $\mathbf{G}(u, v)$  sera donné par :

$$\mathbf{G}(u, v) = (u, v, 0)^T + \left( \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|}, \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|}, \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} \right)^T.$$

Notons, pour simplifier,

$$(e_1, e_2, e_3)^T = \left( \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|}, \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|}, \epsilon(\mathbf{m}(u, v)) \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} \right)^T$$

où  $e_1, e_2$  et  $e_3$  sont des fonctions de  $u$  et  $v$ , alors :

$$\mathbf{G}(u, v) = \begin{pmatrix} u + e_1 \\ v + e_2 \\ e_3 \end{pmatrix}$$

et ses dérivées par rapport à  $u$  et  $v$  sont :

$$\mathbf{G}_u(u, v) = \frac{\partial \mathbf{G}}{\partial u} = \begin{pmatrix} 1 + \frac{\partial e_1}{\partial u} \\ \frac{\partial e_2}{\partial u} \\ \frac{\partial e_3}{\partial u} \end{pmatrix} = \begin{pmatrix} 1 + e_{1u} \\ e_{2u} \\ e_{3u} \end{pmatrix} \quad (5.9)$$

$$\text{et } \mathbf{G}_v(u, v) = \frac{\partial \mathbf{G}}{\partial v} = \begin{pmatrix} e_{1v} \\ 1 + e_{2v} \\ e_{3v} \end{pmatrix}. \quad (5.10)$$

On a  $\mathbf{n}_G = \mathbf{G}_u \times \mathbf{G}_v$ .

Et à la limite (*i.e.*, quand  $\delta u, \delta v \rightarrow 0$ ),

$$\mathbf{n}_F = (0, 0, 1)^T,$$

et si  $\varphi$  est l'angle entre les deux normales  $\mathbf{n}_G$  et  $\mathbf{n}_F$ ,

$$\cos \varphi = \frac{\mathbf{n}_G \cdot \mathbf{n}_F}{\|\mathbf{n}_G\| \cdot \|\mathbf{n}_F\|},$$

soit

$$\cos \varphi = \frac{(\mathbf{G}_u \times \mathbf{G}_v) \cdot (0, 0, 1)^T}{\|\mathbf{G}_u \times \mathbf{G}_v\|}.$$

Et, en utilisant les expressions (5.9) et (5.10) des dérivées de  $\mathbf{G}$  ainsi que le calcul de  $\mathbf{G}_u \times \mathbf{G}_v$  dans [5], on a :

$$\mathbf{G}_u \times \mathbf{G}_v = \begin{pmatrix} 1 + e_{1u} \\ e_{2u} \\ e_{3u} \end{pmatrix} \times \begin{pmatrix} e_{1v} \\ 1 + e_{2v} \\ e_{3v} \end{pmatrix} = \begin{pmatrix} e_{2u}e_{3v} - e_{3u}e_{2v} - e_{3u} \\ e_{3u}e_{1v} - e_{3v}e_{1u} - e_{3v} \\ 1 + e_{1u} + e_{2v} + e_{1u}e_{2v} - e_{2u}e_{1v} \end{pmatrix}$$

et

$$\begin{aligned} \|\mathbf{G}_u \times \mathbf{G}_v\| &= ((e_{2u}e_{3v} - e_{3u}e_{2v} - e_{3u})^2 + (e_{3u}e_{1v} - e_{3v}e_{1u} - e_{3v})^2 \\ &\quad + (1 + e_{1u} + e_{2v} + e_{1u}e_{2v} - e_{2u}e_{1v})^2)^{\frac{1}{2}} \\ &= 1 + e_{1u}^2 + e_{2v}^2 + e_{3u}^2 + e_{3v}^2 + 2e_{1u} + 2e_{2v} + 4e_{1u}e_{2v} - 2e_{2u}e_{1v} + \dots \end{aligned}$$

Notons  $A = e_{1u}^2 + e_{2v}^2 + e_{3u}^2 + e_{3v}^2 + 2e_{1u} + 2e_{2v} + 4e_{1u}e_{2v} - 2e_{2u}e_{1v}$ , on a alors :

$$\|\mathbf{G}_u \times \mathbf{G}_v\|^{-1} = (1 + A)^{-\frac{1}{2}} = 1 - \frac{A}{2} + \frac{3}{8}A^2 + \dots,$$

soit

$$\|\mathbf{G}_u \times \mathbf{G}_v\|^{-1} = 1 + e_{1u}^2 + e_{2v}^2 - \frac{e_{3u}^2}{2} - \frac{e_{3v}^2}{2} - (e_{1u} + e_{2v}) + e_{1u}e_{2v} + e_{2u}e_{1v}.$$

Ainsi,

$$\begin{aligned} \frac{(\mathbf{G}_u \times \mathbf{G}_v) \cdot (0, 0, 1)^T}{\|\mathbf{G}_u \times \mathbf{G}_v\|} &= (1 + e_{1u} + e_{2v} + e_{1u}e_{2v} - e_{2u}e_{1v}) (1 + e_{1u}^2 + e_{2v}^2 \\ &\quad - \frac{e_{3u}^2}{2} - \frac{e_{3v}^2}{2} - (e_{1u} + e_{2v}) + e_{1u}e_{2v} + e_{2u}e_{1v}) \\ &= 1 - \frac{(e_{3u})^2}{2} - \frac{(e_{3v})^2}{2}. \end{aligned}$$

En conclusion,

$$\cos \varphi \approx 1 - \frac{(e_{3u})^2}{2} - \frac{(e_{3v})^2}{2}. \quad (5.11)$$

Calculons alors les dérivées  $e_{3u}$  et  $e_{3v}$ . D'après ce qui précède :

$$e_3(u, v) = \boldsymbol{\epsilon}(\mathbf{m}(u, v)) \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} = \boldsymbol{\epsilon}(\mathbf{m}(u, v)) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \epsilon_3(\mathbf{m}(u, v)),$$

donc

$$e_{3u} = \frac{\partial e_3}{\partial u} = \frac{\partial \epsilon_3(m_1, m_2)}{\partial u} \quad \text{et} \quad e_{3v} = \frac{\partial e_3}{\partial v} = \frac{\partial \epsilon_3(m_1, m_2)}{\partial v}.$$

D'autre part,

$$\frac{\partial \epsilon_3}{\partial u} = \frac{\partial \epsilon_3}{\partial m_1} \cdot \frac{\partial m_1}{\partial u} + \frac{\partial \epsilon_3}{\partial m_2} \cdot \frac{\partial m_2}{\partial u} \quad \text{et} \quad \frac{\partial \epsilon_3}{\partial v} = \frac{\partial \epsilon_3}{\partial m_1} \cdot \frac{\partial m_1}{\partial v} + \frac{\partial \epsilon_3}{\partial m_2} \cdot \frac{\partial m_2}{\partial v}$$

et d'après l'équation (5.8),

$$\begin{aligned} \frac{\partial m_1}{\partial u} &= \frac{1}{\|\mathbf{F}_u\|} ; & \frac{\partial m_2}{\partial u} &= 0 \\ \frac{\partial m_1}{\partial v} &= \frac{-\cos \theta}{\sin \theta \cdot \|\mathbf{F}_v\|} ; & \frac{\partial m_2}{\partial v} &= \frac{1}{\sin \theta \cdot \|\mathbf{F}_v\|} \end{aligned}$$

alors, les deux dérivées  $e_{3u}$  et  $e_{3v}$  sont définies par :

$$e_{3u} = \frac{\partial \epsilon_3}{\partial u} = \frac{\partial \epsilon_3}{\partial m_1} \cdot \frac{1}{\|\mathbf{F}_u\|}, \quad (5.12)$$

$$e_{3v} = \frac{\partial \epsilon_3}{\partial v} = \frac{\partial \epsilon_3}{\partial m_1} \cdot \frac{-\cos \theta}{\sin \theta} \cdot \frac{1}{\|\mathbf{F}_u\|} + \frac{\partial \epsilon_3}{\partial m_2} \cdot \frac{1}{\sin \theta \cdot \|\mathbf{F}_v\|} \quad (5.13)$$



Considérons l'équation (5.11) et, pour alléger l'écriture, notons  $X$  son second membre :

$$X = 1 - \frac{(e_{3u})^2}{2} - \frac{(e_{3v})^2}{2}.$$

En utilisant les expressions (5.12) et (5.13), on a :

$$\begin{aligned} X &= 1 - \frac{1}{2} \left( \frac{\partial \epsilon_3}{\partial m_1} \right)^2 \cdot \frac{1}{\|\mathbf{F}_u\|^2} \\ &\quad - \frac{1}{2} \left[ \frac{\partial \epsilon_3}{\partial m_1} \cdot \frac{-\cos \theta}{\sin \theta} \cdot \frac{1}{\|\mathbf{F}_u\|} + \frac{\partial \epsilon_3}{\partial m_2} \cdot \frac{1}{\sin \theta \cdot \|\mathbf{F}_v\|} \right]^2 \\ &= 1 - \frac{1}{2} \left( \frac{\partial \epsilon_3}{\partial m_1} \right)^2 \cdot \frac{1}{\|\mathbf{F}_u\|^2} \\ &\quad - \frac{1}{2} \left[ \left( \frac{\partial \epsilon_3}{\partial m_1} \right)^2 \cdot \frac{\cos^2 \theta}{\sin^2 \theta \cdot \|\mathbf{F}_u\|^2} + \left( \frac{\partial \epsilon_3}{\partial m_2} \right)^2 \cdot \frac{1}{\sin^2 \theta \cdot \|\mathbf{F}_v\|^2} \right. \\ &\quad \left. - 2 \frac{\partial \epsilon_3}{\partial m_1} \cdot \frac{\partial \epsilon_3}{\partial m_2} \cdot \frac{\cos \theta}{\sin^2 \theta \cdot \|\mathbf{F}_u\| \cdot \|\mathbf{F}_v\|} \right] \\ &= 1 - \frac{1}{2} \left( \frac{\partial \epsilon_3}{\partial m_1} \right)^2 \cdot \frac{1}{\|\mathbf{F}_u\|^2} \\ &\quad - \frac{1}{2} \left( \frac{\partial \epsilon_3}{\partial m_1} \right)^2 \cdot \frac{\cos^2 \theta}{\sin^2 \theta \cdot \|\mathbf{F}_u\|^2} - \frac{1}{2} \left( \frac{\partial \epsilon_3}{\partial m_2} \right)^2 \cdot \frac{1}{\sin^2 \theta \cdot \|\mathbf{F}_v\|^2} \\ &\quad + \frac{\partial \epsilon_3}{\partial m_1} \cdot \frac{\partial \epsilon_3}{\partial m_2} \cdot \frac{\cos \theta}{\sin^2 \theta \cdot \|\mathbf{F}_u\| \cdot \|\mathbf{F}_v\|}. \end{aligned}$$

Comme  $m_i$  est une fonction qui envoie les paramètres  $u, v$  du plan tangent dans le domaine paramétrique  $D$ , donc  $m_i(u, v)$  est dans le domaine  $D$ , on peut alors utiliser la constante de Lipschitz  $L$  définie au chapitre 3.1.1, soit

$$\left| \frac{\partial \epsilon_3}{\partial m_i} \right| \leq L, \quad i = 1, 2.$$

Par conséquent,

$$- \left( \frac{\partial \epsilon_3}{\partial m_i} \right)^2 \geq -L^2$$

et

$$\left| \frac{\partial \epsilon_3}{\partial m_1} \right| \cdot \left| \frac{\partial \epsilon_3}{\partial m_2} \right| = \left| \frac{\partial \epsilon_3}{\partial m_1} \frac{\partial \epsilon_3}{\partial m_2} \right| \leq L^2 \Rightarrow -L^2 \leq \frac{\partial \epsilon_3}{\partial m_1} \frac{\partial \epsilon_3}{\partial m_2}.$$

On peut maintenant donner une première minoration de  $X$  que nous noterons  $X_1$ .

$$\begin{aligned}
X &\geq X_1 = 1 - \frac{1}{2} L^2 \cdot \frac{1}{\|\mathbf{F}_u\|^2} \\
&\quad - \frac{1}{2} L^2 \cdot \frac{\cos^2 \theta}{\sin^2 \theta \cdot \|\mathbf{F}_u\|^2} - \frac{1}{2} L^2 \cdot \frac{1}{\sin^2 \theta \cdot \|\mathbf{F}_v\|^2} \\
&\quad - L^2 \cdot \frac{\cos \theta}{\sin^2 \theta \cdot \|\mathbf{F}_u\| \cdot \|\mathbf{F}_v\|} \\
X_1 &= 1 - L^2 \left[ \frac{1}{2 \|\mathbf{F}_u\|^2} \left( 1 + \frac{\cos^2 \theta}{\sin^2 \theta} \right) \right. \\
&\quad \left. + \frac{1}{2 \sin^2 \theta \cdot \|\mathbf{F}_v\|^2} + \frac{\cos \theta}{\sin^2 \theta \cdot \|\mathbf{F}_u\| \cdot \|\mathbf{F}_v\|} \right]
\end{aligned}$$

Soit, en remplaçant dans l'équation (5.11),

$$\cos \varphi \geq X_1 = 1 - \frac{L^2}{2} \left[ \frac{\|\mathbf{F}_u\|^2 + \|\mathbf{F}_v\|^2 + 2 \cos \theta \cdot \|\mathbf{F}_u\| \cdot \|\mathbf{F}_v\|}{\sin^2 \theta \cdot \|\mathbf{F}_u\|^2 \cdot \|\mathbf{F}_v\|^2} \right]. \quad (5.14)$$

Introduisons maintenant les coefficients  $E, F$  et  $G$  de la première forme fondamentale de  $\mathbf{F} = \mathbf{F}(u, v)$  définis au début de cette section. De (5.3), nous concluons :

$$\begin{aligned}
\cos^2 \theta &= \frac{(\mathbf{F}_u \cdot \mathbf{F}_v)^2}{\|\mathbf{F}_u\|^2 \cdot \|\mathbf{F}_v\|^2} = \frac{F^2}{EG} \\
\sin^2 \theta = 1 - \cos^2 \theta &= 1 - \frac{F^2}{EG} = \frac{EG - F^2}{EG}.
\end{aligned} \quad (5.15)$$

Notons que  $EG - F^2 = \|\mathbf{F}_u \cdot \mathbf{F}_v\|^2$  est non nul, nous pouvons donc réécrire l'équation (5.14) de la manière suivante :

$$\cos \varphi \geq 1 - \frac{L^2}{2} \left[ \frac{E + 2F + G}{EG - F^2} \right]$$

□

**Corollaire 5.2.1.** *Si  $\mathbf{F}$  est une patch donnée représentée par un réseau d'isoparamétriques orthogonales et  $\mathbf{G}$  son extension de Whitney, alors l'angle  $\varphi$  entre les normales de ces deux surfaces vérifie*

$$\cos \varphi \geq 1 - \frac{L^2}{2} \left[ \frac{E + G}{EG} \right].$$

*Et si, de plus, les vecteurs tangents à  $\mathbf{F}$  sont unitaires, alors*

$$\cos \varphi \geq 1 - L^2.$$

*Démonstration du corollaire*

Si  $\mathbf{F}$  est représentée par un réseau d'isoparamétriques orthogonales alors  $\theta = \frac{\pi}{2}$  et par suite  $\cos \theta = 0$ , c'est-à-dire  $F = 0$ . En remplaçant dans le théorème 5.2.1, on a directement

$$\cos \varphi \geq 1 - \frac{L^2}{2} \left[ \frac{E+G}{EG} \right].$$

Et si, de plus, les vecteurs tangents à  $\mathbf{F}$  sont unitaires, on a  $\|\mathbf{F}_u\|^2 = \|\mathbf{F}_v\|^2 = 1$ , i.e.,  $E = G = 1$ , d'où la deuxième inégalité.  $\square$

Posons  $K^2 = \frac{1}{2} \left[ \frac{E+2F+G}{EG-F^2} \right]$ , ( $K$  est fonction des dérivées de  $\mathbf{F}$  au point  $(u, v)$ ) alors

$$1 - (LK)^2 \leq |\cos \varphi| = 1 - \frac{\varphi^2}{2} + \dots \iff |\varphi| \leq \sqrt{2}L|K|.$$

**Exemple 5.2.1.** Reprenons l'exemple de la section 3.1.3. On a

$$\mathbf{F}(u, v) = Id(u, v) \quad \text{alors} \quad \mathbf{F}_u^2 = \nabla_u \mathbf{G}^2 = 1, \quad \text{i.e.,} \quad E = G = 1$$

et

$$\mathbf{F}_u \cdot \mathbf{F}_v = 0, \quad \text{i.e.,} \quad F = 0.$$

Alors  $K = 1$  et par suite

$$\cos \varphi \geq 1 - L^2.$$

Et on retrouve le résultat du corollaire pour le cas particulier traité dans l'exemple.

Comme dans cet exemple,  $L = 0.01114$  alors  $|\varphi| \leq 0.01575$  radians.

### 5.3 Non-intersection des faces du *QuasiNURBS*

Pour la détection de non-autointersection, nous nous basons sur le critère de Volino-Thalman décrit dans le chapitre 4. Nous devons vérifier deux hypothèses :

1. l'existence d'un vecteur  $\mathbf{v}$  tel que, pour toute normale  $\mathbf{n}$  de la patch considérée,  $\mathbf{n} \cdot \mathbf{v} > 0$ ,
2. la projection du contour de la *patch* ne présente pas d'autointersection.

Rappelons que la *patch* considérée ici est la *patch* perturbée  $\mathbf{G}$ .

Prouver l'existence d'un vecteur  $\mathbf{v}$  vérifiant la première condition revient à montrer que l'ensemble des normales de  $\mathbf{G}$  est compris dans un cône dont le demi-angle est inférieur à  $\frac{\pi}{2}$ . Comme nous avons l'information sur les normales de  $\mathbf{F}$ , nous déterminerons le cône des normales de  $\mathbf{G}$  à partir de cette information.

Nous avons vu dans la section précédente que l'angle  $\varphi$  entre les normales  $\mathbf{n}_G$  et  $\mathbf{n}_F$  de  $\mathbf{G}$  et  $\mathbf{F}$  respectivement est majoré par  $\sqrt{2}L|K|$ . Comme  $K$  est liée aux dérivées de  $\mathbf{F}$  au point  $(u, v)$  et  $L$  aux dérivées de  $\epsilon$ , si, pour une valeur donnée des paramètres  $(u, v)$ , les dérivées de  $\mathbf{F}$  et de  $\epsilon$  sont telles que  $\sqrt{2}L|K|$  est très petit, l'angle  $\varphi$  est très petit aussi. Dans ce cas et à condition que le demi-angle du cône des normales de  $\mathbf{F}$  ne soit pas très proche de  $\frac{\pi}{2}$  (tout en étant inférieur à  $\frac{\pi}{2}$ ), le cône des normales de  $\mathbf{G}$  est de demi-angle inférieur à  $\frac{\pi}{2}$ . Il existera donc un vecteur  $\mathbf{v}$  tel que  $\mathbf{n}_G \cdot \mathbf{v} > 0$ , pour toute normale  $\mathbf{n}_G$  de  $\mathbf{G}$ . Et la première condition sera vérifiée.

Pour la deuxième condition, nous devons nous assurer que la projection de la frontière de la *patch* perturbée  $\mathbf{G}$  ne présente pas d'autointersection. Concernant les frontières de  $\mathbf{G}$ , nous avons établi dans la section 3.1.1 que les applications  $\mathbf{F}(\mathbf{p}^k(t)) \rightarrow \beta^k(t)$ , de l'ensemble image  $\text{Im}(\mathbf{F}(\mathbf{p}^k(t)))$  vers l'ensemble image  $\text{Im}(\beta^k(t))$ , sont bijectives donc qu'il y a une correspondance point par point entre chacune des courbes  $\mathbf{F}(\mathbf{p}^k(t))$  (frontières de la *trimmed patch* donnée) et les courbes  $\beta^k(t)$  (frontières de la *trimmed patch* perturbée). Cependant, si la projection des frontières de  $\mathbf{F}$  selon le vecteur  $\mathbf{v}$  ne présente pas d'autointersection, on ne peut rien en conclure quant à la projection des frontières de  $\mathbf{G}$ . Il y a en plus le problème non trivial de trouver une paramétrisation qui nous permettra de définir une frontière commune aux *trimmed patches* adjacentes. Et c'est pour cela que la deuxième condition du critère de non-autointersection de Volino-Thalmann est souvent ignorée [29].

Dans le cas de deux *trimmed patches* adjacentes de l'ensemble *QuasiNURBS*, la seule intersection de ces deux *trimmed patches* devrait être, selon les données topologiques, une courbe. Pour prouver que ces deux *trimmed patches* n'ont pas

d'autres intersections, nous considérerons qu'ils sont les composantes connexes d'une même *patch*. Le critère de non-autointersection, décrit dans la section précédente sera alors appliqué à la *patch* composée.

La question ici est : comment faire pour que les deux *trimmed patches* décrites dans des domaines paramétriques différents, soient les composantes d'une même *patch*?

Pour deux *trimmed NURBS patches* adjacentes, on devra définir une transformation qui amènera les deux domaines paramétriques dans un seul et se rappeler que les deux *trimmed patches* n'ont pas la même valeur pour  $L$ . Cela revient à trouver une nouvelle paramétrisation pour chaque *trimmed NURBS patch* de manière à faire coïncider la courbe d'intersection dans un même domaine paramétrique.

Mais pour deux *trimmed patches* de subdivision, les choses se compliquent car au lieu de domaines paramétriques, les préimages des limites des *trimmed patches* sont définies dans deux polyèdres distincts de  $R^3$ . Considérer alors les deux *trimmed patches* comme composantes d'une seule surface n'est pas évident. Donc, dire [29] qu'on peut "simplement" appliquer le test de détection de l'autointersection à l'union des deux *patches*, est un peu trop rapide. En effet, l'application serait simple une fois l'union faite et si on ne considère pas des *trimmed patches*. Et l'union des deux *trimmed patches* nécessite d'abord de déterminer un polyèdre commun où seront définies les frontières de deux *patches* adjacentes. Ce qui n'est pas du tout simple.

# Chapitre 6

## Conclusion

### 6.1 Conclusion de notre travail

Le problème de robustesse en modélisation des solides est très complexe et il implique plusieurs domaines des mathématiques et de l'informatique. Comme nous l'avons montré plus haut, les différentes méthodes présentées, tout en y apportant une contribution intéressante pour la plupart, ont échoué à le résoudre réellement.

Plusieurs sources d'erreurs, inévitables, en sont la principale raison. La première que nous avons considérée, est l'incertitude sur les données. Cette erreur est souvent négligée. Et pourtant, nous devons en tenir compte car il est nécessaire de bien définir le solide que nous présentons au problème. Nous pensons que la résolution de cette première partie du problème permettra de construire une théorie rigoureuse vers des opérations géométriques robustes entre solides. En effet, l'incertitude sur les données peut rendre la représentation des solides, fournie au problème, tout à fait incohérente ; elle peut donc ne représenter aucun solide.

Hoffmann a posé ce problème à travers l'énoncé de ce que serait une opération correcte et nous, nous allons plus loin en définissant exactement ce que veut dire “le modèle” cité par Hoffmann et que nous avons appelé “réalisation”.

Nous devons préciser qu'une opération géométrique robuste avec des données incertaines n'est pas nécessairement une opération très précise mais une opération qui donne un résultat valide, solution d'un problème aussi proche que possible du

problème posé.

Ainsi, la première partie de notre travail a été la proposition d’une définition rigoureuse de l’ensemble que représente ces données et que nous avons appelé “*QuasiNURBS*”. Un ensemble *QuasiNURBS* est un ensemble compact régulier ou r-ensemble dont les faces frontières sont une perturbation des données géométriques (NURBS). Cet ensemble est une réalisation dans  $R^3$  de la représentation donnée. Basées sur le théorème d’extension de Whitney, des conditions suffisantes de l’existence d’une telle réalisation (pour une représentation donnée) ont été présentées ainsi que les bornes sur la distance entre les données et les frontières des ensembles *QuasiNURBS* correspondantes. On a ainsi démontré que la perturbation appliquée, n’est nulle part plus grande que celle déjà existante dans les données concernant les frontières de la *patch*. En plus, les *patches* perturbées vérifient la condition de Lipschitz à l’intérieur de leur domaine de définition. Un exemple d’intersection de deux solides simples a illustré le problème pour l’intersection de trois faces et décrit la solution que nous avons proposé. Cette première partie de notre travail a été présentée dans [4]. On a aussi montré, dans [74] que la même démarche est possible pour les *trimmed patches* de subdivision combinée.

Le théorème de Whitney nous permet de garantir une jonction exacte entre deux faces de l’ensemble *QuasiNURBS* mais ne garantit pas une absence d’auto-intersections indésirables dans une *patch* ou d’intersections parasites entre deux *patches* adjacentes. Pour cette deuxième propriété, il a fallu choisir un critère de détection d’auto-intersection. Le plus adéquat, car plus général, est le critère de Volino-Thalmann mais ce critère n’est pas formulé rigoureusement et a donc laissé place à beaucoup d’ambiguïté quant à son utilisation. Nous avons donc donné des contre-exemples pour montrer que les hypothèses énoncées dans la conjecture ne sont pas suffisantes et que d’autres propriétés doivent être ajoutées. Ce travail fait partie du deuxième papier soumis [5].

Une fois le critère de détection d’auto-intersection énoncé de manière rigoureuse, nous avons concentré notre étude sur les variations des normales du quasi-interpolant décrivant une face de l’ensemble *QuasiNURBS*. En effet, la variation

des vecteurs normaux est une condition importante du critère de détection d'auto-intersection, il est donc important de borner cette variation afin de s'assurer que la perturbation des normales est maîtrisée et appliquer un critère de détection. Et comme chaque face de cet ensemble vérifie la condition de Lipschitz sur tout son domaine, il a été possible de borner la variation de la normale à travers cette *patch* par rapport à la *trimmed patch* donnée. Cette dernière étude paraît dans [74].

Et avec cette dernière partie, nous avons complété la définition d'un ensemble valide. Nous pensons ainsi avoir apporté une contribution à la première partie de la définition d'un algorithme robuste dans une opération géométrique entre solides qui permettra de passer à la deuxième partie de la définition d'un algorithme robuste qui serait l'analyse de l'erreur entre les deux problèmes : le problème posé et le problème réellement résolu.

## 6.2 Travaux futurs

### 6.2.1 Principe de l'analyse inverse de l'erreur

Pour pouvoir exprimer clairement les problèmes ouverts, nous donnons un bref résumé de l'analyse inverse de l'erreur dans le domaine de la modélisation des solides [35, 4].

Le but ultime de l'analyse de l'erreur est de montrer que la solution du problème que nous avons calculée, est en fait la solution légèrement perturbée d'un problème légèrement perturbé [38]. Un problème peut avoir plusieurs solutions si on ne connaît pas la topologie exacte du résultat, le choix parmi ces solutions serait une étude intéressante mais dans le cas de l'analyse inverse de l'erreur, nous nous intéressons à l'aspect numérique du problème.

Pour illustrer cette analyse, considérons l'intersection régularisée entre deux solides  $S_0$  et  $S_1$  soit le calcul de  $S_0 \cap S_1$ . Nous avons défini dans la section 3.1 l'ensemble *QuasiNURBS*, perturbation d'un solide donné. Notons  $S'_0$  et  $S'_1$  les ensembles *QuasiNURBS* correspondant respectivement à  $S_0$  et  $S_1$ . Si on note  $S_c$



le résultat, on doit montrer que :

- i. il existe  $S'_0$ ,
- ii. il existe  $S'_1$

tels que les trois conditions suivantes soient vérifiées :

- 1.  $d(S'_0, S_0)$  est petite,
- 2.  $d(S'_1, S_1)$  est petite,
- 3.  $d(S'_0 \cap^* S'_1, S_c)$  est petite,

où  $d$  est une métrique, par exemple celle définie dans [35].

En fait, les données présentées à l'algorithme sont les représentations décrivant les solides  $S_0$  et  $S_1$  et le résultat du calcul est une représentation de  $S_c$ . Il faut donc d'abord s'assurer que les représentations sont valides et que les solides donnés sont les réalisations de ces représentations. En plus, la distance entre deux solides  $S_0$  et  $S_1$  valides sera déterminée en fonction de leur représentation  $\Delta_0$  et  $\Delta_1$ .

Supposons donc donnés  $(S_0, \Delta_0)$  et  $(S_1, \Delta_1)$  tels que  $S_0 \models \Delta_0$  et  $S_1 \models \Delta_1$ , et que  $\Delta_c$  sont les données représentant le résultat. Alors, nous devons montrer que :

- i. il existe  $S'_0$  et  $\Delta'_0$  tels que  $S'_0 \models \Delta'_0$ ,
- ii. il existe  $S'_1$  et  $\Delta'_1$  tels que  $S'_1 \models \Delta'_1$ ,
- iii. il existe  $S_c$  tel que  $S_c \models \Delta_c$ , et
- iv. il existe  $\Delta_{\cap^*}$  tel que  $S'_0 \cap^* S'_1 \models \Delta_{\cap^*}$

tels que les trois conditions suivantes soient vérifiées :

- 1.  $d(S'_0, S_0)$  est petite,
- 2.  $d(S'_1, S_1)$  est petite,
- 3.  $d(S'_0 \cap^* S'_1, S_c)$  est petite.

Dans l'exemple que nous avons traité dans la section 3.1.3, ce schéma est simplifié si bien que nous avons directement la représentation de l'intersection, c'est à dire que  $S'_0 \cap^* S'_1 = S_c$ . Ce n'est pas le cas en général.

### 6.2.2 Problèmes ouverts

Dès que les calculs se font en précision finie, la question de stabilité numérique [14] doit être étudiée. Même si la méthode numérique est stable, rien ne garantit que l'estimation de l'erreur entre la solution exacte et la solution calculée soit petite; un bon conditionnement du problème est nécessaire. En effet, comme les données de départ sont incohérentes, la méthode numérique donne la solution numérique d'un problème dont les données sont légèrement perturbées (pour garantir la cohérence entre données). Cependant, si le problème est mal conditionné (*i.e.* une petite perturbation des données change complètement la vraie solution, par exemple, à cause du fait que l'angle entre deux faces des objets considérés dans l'opération géométrique, est très petit), il ne supportera pas les perturbations, même petites.

Des données incertaines dans ce problème sont donc une motivation pour une analyse inverse de l'erreur [35] : il faut démontrer que la solution obtenue est solution d'un problème proche du problème réel mais fort probablement non identique. Et comme les autres sources d'erreurs sont inévitables, cela nous donne une solution aussi bonne qu'on puisse espérer. Ainsi, si l'erreur inverse est plus petite que l'incertitude sur les données alors la solution calculée peut difficilement être critiquée.

Donc, si le problème est mal conditionné, on admet qu'on ne pourra rien faire au niveau de l'erreur qui en découle. Cependant, il serait intéressant d'étudier séparément l'effet de cette erreur même si on ne peut pas l'éviter. C'est une issue à investir.

Et comme le (ou les) objet(s) soumis à une opération donnée subissent des perturbations dues à l'arithmétique en virgule flottante, il serait donc important d'analyser l'erreur provoquée par ses perturbations (troncature, arrondi). En effet, le résultat de l'opération sera la solution exacte d'un problème perturbé (par les calculs machines), il faudra alors démontrer que ce problème est proche, selon une distance bien choisie [35], du problème initial. Même si les données incertaines en entrée motivent une analyse inverse (vu que nous étudions l'erreur dans l'espace

problème), ceci n'empêchera pas une analyse de l'erreur dans l'espace solution.

Parmi les problèmes ouverts qui suivraient ce travail, il y aurait donc :

1. le conditionnement du problème géométrique,
2. la métrique pour mesurer la distance entre objets,
3. et le plus important problème : la stabilité de l'algorithme géométrique, c'est-à-dire l'analyse inverse de l'erreur.

On soulignera que ce qui sera soumis à l'opération, sera la représentation  $\Delta$  mais ce sont les face perturbées  $\mathbf{F} + \epsilon$  sur lesquelles devra se faire l'analyse inverse puisque la représentation  $\Delta$  n'est pas cohérente et ne permet donc pas d'établir des théorèmes qui font référence à des sous-ensembles de  $R^3$ .

# Bibliographie

- [1] ACIS 3D Toolkit, Spatial Technology, Boulder, CO, 1998.
- [2] L.-E. Andersson, T. J. Peters and N. F. Stewart, “Selfintersection of composite curves and surfaces”. *Computer Aided Geometric Design* 15, No.5, 507-527, 1998.
- [3] G. Aronsson, “Extension of functions satisfying Lipschitz conditions”. *Arkiv för Matematik* (6), No. 28, Stockholm, 1967.
- [4] L.-E. Andersson, N. F. Stewart and M. Zidani, “Error analysis for operations in solid modeling in the presence of uncertainty”, Manuscript, 2004.
- [5] L.-E. Andersson, N. F. Stewart and M. Zidani, “Conditions for use of a non-selfintersection conjecture”, Manuscript, 2004.
- [6] W. Böhm, G. Farin and J. Kahmann, “A survey of curve and surface methods in CAGD”. *CAGD* 1, 1-60, 1984.
- [7] G. Birkhoff, W. J. Gordon, “The draftsman’s and related equations”. *J. Approximation Theory* 1, 199-208, 1968.
- [8] G.-Z. Chang and P. J. Davis, “The convexity of Bernstein Polynomials over triangles”. *J. of Approximation Theory* 40, 11-28, 1984.
- [9] M. S. Casale, J. E. Bobrow, “A set operation algorithm for sculptured solids modeled with trimmed patches”. *Computer Aided Geometric Design* 6, No.3, 235 - 247, August 1989.
- [10] E. Catmull and J. Clark, “Recursively generated B-spline surfaces on arbitrary topological meshes”. *CAD* Vol.10, No.6, 350-355, 1978.

- [11] S. A. Coons, "Surfaces for computer aided design of space forms". Rep. MAC-TR-41, Project MAC, MIT, Cambridge, Mass., 1967.
- [12] M. Daniel, "Using a convex pyramid to bound surface normal vectors". *Computer Graphics Forum* 15, 4, 219-227, 1996.
- [13] K. Deimling, Non-linear Functional Analysis. Springer-Verlag, 1984.
- [14] P. Deuffhard and A. Hohmann, "Numerical Analysis in Modern Scientific Computation. An Introduction". Second Edition, Springer-Verlag, New York, 2003
- [15] M. P. do Carmo, "Differential Geometry of Curves and Surfaces". Prentice-Hall, 1976.
- [16] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points". *CAD* Vol.10, No.6, 356-360, Nov.1978.
- [17] H. Desaulniers and N. F. Stewart, "Quasi-rectilinear r-sets". *Proceedings of the Fourth Canadian Conference on Computational Geometry*, 52-58, August 10-14, 1992.
- [18] H. Desaulniers and N. F. Stewart, "Backward error analysis for floating-point operations on rectilinear r-sets". Publication #816, Département d'informatique et de recherche opérationnelle, Université de Montréal, Janvier 1993.
- [19] H. Desaulniers and N. F. Stewart, "Robustness of numerical methods in geometric computation when problem data is uncertain". *CAD* Vol.25, No.9, 539-545, Sept. 1993.
- [20] DT\_NURBS Spline Geometry Subprogram Library User's Manual, Version 3.6. Carderock Division Naval Surface Warfare Center, Bethesda, Maryland, 20084-5000, CARDEROCKIV-94/000, October 1998.
- [21] H. Edelsbrunner and E. Mücke, "Simulation of Simplicity : A technique to cope with degenerate cases in geometric algorithms". *Proc. Fourth ACM Symp. on Computer Geometry*, 118-133, June 1988.
- [22] S. Fang and B. Bruderlin, "Robustness in geometric modeling-Tolerance-based methods". *Lecture notes in computer science 553, Computational geo-*

- metry Methods, Algorithms and Applications*, Bieri and Noltemeier (Eds), 85-101, 1991.
- [23] G. Farin, D. Hansford, "Discrete Coons patches". *CAGD* 16, 691-700, 1999.
- [24] R. T. Farouki, "Closing the gap between CAD Model and downstream application". *SIAM News* (32), No.1, June 1999.
- [25] R. T. Farouki and V. T. Rajan, "Algorithms for polynomials in Bernstein form". *CAGD* 5, 1-26, 1988.
- [26] W. J. Gordon, "Spline-Blended Surface Interpolation Through Curve Networks". *Journal of Mathematics and Mechanics* Vol.18, No.10, 1969.
- [27] R. N. Goldman, "Subdivision algorithms for Bézier triangles". *CAD* Vol.15, No.3, Mai 83.
- [28] T. A. Grandine and F. W. Klein, "A new approach to the surface intersection problem". *Computer Aided Geometric Design* 14, 111-114, 1997.
- [29] E. Grinspun and P. Schröder, "Normal bounds for subdivision-surface interference detection". *Proceedings of the IEEE Conference on Visualisation*, 333-340, Oct.2001.
- [30] L. Gross and G. Farin, "A transfinite form of Sibson's interpolant". *Discrete Applied Mathematics* (93), 33-50, 1999.
- [31] C. M. Hoffmann, J. E. Hopcroft and M. S. Karasick, "Towards implementing robust geometric computations". *Proceedings of the Fourth Annual ACM Conference on Computational Geometry*, Urbana-Champaign, IL, 106-117, 1988.
- [32] C. M. Hoffmann, "The problems of accuracy and robustness in geometric computations". *IEEE Computer*, 31-41, Mars 1989.
- [33] C. M. Hoffmann, "Geometric and Solid Modeling : An Introduction". Morgan Kaufmann Publishers, Inc., San Mateo, California, 1989.
- [34] C. M. Hoffmann, "Robustness in geometric computations". *J. Computing and Information Science in Engineering* (1), 143-156, 2001.

- [35] C. M. Hoffmann and N. F. Stewart, “Accuracy and semantics in shape-interrogation applications”. *Graphical Models* 67, No.5, 373-389, Sept. 2005.
- [36] M. E. Hohmeyer, “A surface intersection algorithm based on loop detection”. *Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications* , Austin, Texas, United States, 197-207, June 05-07, 1991.
- [37] J. E. Hopcroft and P. J. Kahn, “A paradigm for robust geometric algorithms”. Technical Report TR89-1044, Department of Computer Science, Cornell University, Ithaca, NY, 1989.
- [38] W. M. Kahan, “A survey of error analysis”, *Information Processing '71*, North Holland, 1214-1239, 1971.
- [39] O. Kavian, “Introduction à la Théorie des Points Critiques”. Springer-Verlag, Mathématiques et Applications, 1993.
- [40] L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, and C. Yap, “Classroom Examples of Robustness Problems in Geometric Computations”. *Proc. of the 12th Annu. European Sympos. Algorithms (ESA '04)*, Bergen, Norway. LNCS 3221, Springer, 702-713, September, 2004.
- [41] P. Kiciak, “Computing normal vector Bézier patches”. *CAGD* 18, 699-710, 2001.
- [42] D.-S. Kim, “Cones on Bézier curves and surfaces”. Ph.D. Dissertation, Department of industrial and operations engineering, The University of Michigan, Ann Arbor, MI.
- [43] D.-S. Kim, P. Y. Papalambros and T. C. Woo, “Tangent, normal and visibility cones on Bézier surfaces”. *CAGD* 12, 305-320, 1995
- [44] V. Kreinovich, “How to take into account dependance between the inputs : from interval computations to constraint-related set computations”. *Dagstuhl Seminar* 06021, 8-13 janvier, 2006.
- [45] S. Krishnan and D. Manocha, “Algebraic Loop Detection and Evaluation Algorithms for Curve and Surface Interrogations”, Graphics Interface '96,

- Canadian Human-Computer Communications Society, Wayne A. Davis and Richard Bartels editors, 87-94, 1996.
- [46] D. Lasser, “Tensor product Bézier surfaces on triangle Bézier surfaces”. *CAGD* 19, 625-643, 2002.
- [47] A. Levin, “Combined subdivision schemes for the design of surfaces satisfying boundary conditions”. *CAGD* 16, 345-354, 1999.
- [48] A. Lieutier, “Une formalisation du besoin d’algorithmes robustes en géométrie”. *Second Real Numbers and Computers Conference*, 9-11 April, 1996.
- [49] M. Lipschutz, “Theory and Problems of Differential Geometry”. Schaum’s outline series, McGraw-Hill, 1969.
- [50] N. Litke, A. Levin and P. Schröder, “Trimming for subdivision surfaces”. *CAGD* 18, 463-481, 2001.
- [51] C. T. Loop, “Smooth subdivision surfaces based on triangles”. Master’s Thesis, Department of Mathematics, University of Utah, 1987.
- [52] E. J. McShane, “Extension of Range of Functions”. *Bull. Amer. Math. Soc.* (40), 837-842, 1934.
- [53] D. Michelucci, “Arithmetic issues in geometric computations”. *2nd conference on Real Numbers and Computers*, Marseille, France, 43-69, Avril 1996.
- [54] V. Milenkovic, “Verifiable implementations of geometric algorithms using finite precision arithmetic”. Techn. report CS-88-168, Carnegie Mellon Univ., Pittsburgh, 1988.
- [55] R. E. Moore, “Interval Analysis”. Prentice-Hall, 1966.
- [56] J. Oprea, “Differential Geometry and Its Applications”. Prentice Hall 2nd ed., 2004.
- [57] N. Patrikalakis and T. Maekawa, “Shape Interrogation for Computer Aided Design and Manufacturing”, Springer-Verlag, 2002.
- [58] T. J. Peters, Private communication.



- [59] L. Piegl and W. Tiller, “The NURBS Book”. Springer-Verlag, 1997.
- [60] A. Pressley, “Elementary Differential Geometry”. Springer, 2001.
- [61] A. A. G. Requicha, “Mathematical models of rigid solid objects”. Technical Memo TM-28, University of Rochester, 1977.
- [62] A. A. G. Requicha, “Representations for rigid solids : theory, methods and systems”, *ACM Computing Surveys* (12), No.4, 437-464, 1980.
- [63] A. A. G. Requicha and J. R. Rossignac, “Solid Modeling and Beyond”. *IEEE Computer Graphics and Applications*, 31-44, September 1992.
- [64] T. Saito, G.-J. Wang and T. W. Sederberg, “Hodographs and normals of rational curves and surfaces”. *Computer Aided Geometric Design* 12, 417-430, 1995.
- [65] S. Schirra, “Precision and robustness in geometric computations”. in *Algorithmic Foundations of Geographic Information Systems*, LNCS 1340, Springer-Verlag, 1997.
- [66] T. W. Sederberg and E.-W. Chionh, “The algebra and geometry of curve and surface intersection”. Manuscript, 2002.
- [67] J. R. Shewchuk, “Triangle : Engineering a 2D quality mesh generator and delaunay triangulator”. Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry : Towards Geometric Engineering (WACG96)*, 203-222, 1996.
- [68] T. W. Sederberg and R. J. Meyers, “Loop detection in surface patch intersection”. *CAGD* 5, 161-171, 1988.
- [69] R. Seidel, “The nature and meaning of perturbations in geometric computing”. *Discrete & Computational Geometry* 19, 1-17, 1998.
- [70] M. Seegal and C. H. Séquin, “Consistent Calculations for Solids Modeling”. *ACM Symposium on Computational Geometry*, 29-38, 1985.
- [71] X. Song, T. W. Sederberg, J. Zheng, R. Farouki and J. Hass, “Linear perturbation methods for topologically consistent representations of free-form surface intersections”. *CAGD* 21, 303-321, 2004.

- [72] J. Stam, "Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values". *Proc.ACM SIGGRAPH*, 395-404, 1998.
- [73] STEP International Standard, ISO 10303-42, "Industrial automation systems and integration - Product data representation and exchange - Part 42 : Integrated generic resources : Geometric and topological representation", International Organization for Standardization (ISO), Reference Number ISO 10303-42 : 1994(E), First Edition 1994-12-15, 1997.
- [74] N. F. Stewart and M. Zidani, "Transfinite interpolation for well-definition in error analysis in solid modelling". *Dagstuhl Seminar* 06021, 8-13 janvier, 2006.
- [75] A. J. K. Stewart, "The Theory and Practice of Robust Geometric Computation or How to Build Robust Solid Modelers". Ph.D. Thesis, Department of Computer Science, Cornell University, Ithaca, New York, 91-1229, August 1991.
- [76] P. Volino and N. M. Thalmann, "Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity". *Eurographics'94* Vol.13, No.3, C155-C164, M Daehlen and K. Kjell Dahl, Guest Editors, Blackwell Publishers, 1994.
- [77] H. Whitney, "Analytic Extensions of Differentiable Functions Defined in Closed Sets". *Trans Am. Math. Soc.* (36), 63-89, 1934.
- [78] F. Yamaguchi, "Computer-Aided Geometric Design". Springer-Verlag Tokyo, 2002.
- [79] D. Zorin and P. Schröder, "Subdivision for Modeling and Animation". SIGGRAPH 2000 Course Notes, D.Zorin and P.Schröder Editors.

# Annexe A

## Régularisation

On introduit dans ce qui suit les notions et définitions concernant la description mathématique d'un solide. Ces idées ne sont pas nouvelles mais ce sont des notions de base. Il est nécessaire de définir mathématiquement ce qu'est un solide donc décrire quels sous-ensembles de  $R^3$  sont des modèles valides pour les solides. Une première étape vers une caractérisation de la notion de solidité est d'introduire la notion de régularité [61].

### A.0.3 Régularité des ensembles

Tout sous-ensemble de  $R^3$  ne représente pas nécessairement un solide ; par exemple, un ensemble de points isolés, une ligne ou un plan ne représente aucun solide. Nous ne considérerons donc que des ensembles “réguliers fermés” définis comme suit :

**Définition A.0.1.** *Soit  $X$  un sous-ensemble de  $R^3$ .  $X$  est un ensemble régulier fermé s'il est égal à la fermeture de son intérieur, i.e.,  $X = fe(int(X))$ .*

En fait, un ensemble de points isolés, une ligne ou un plan n'a pas d'intérieur dans la topologie de  $R^3$ . Ils ne peuvent pas, par conséquent, être réguliers fermés. Nous ne considérerons que des ensembles réguliers compacts de  $R^3$  (donc fermés bornés) et semi-analytiques ; l'exemple ci-dessous (FIG. A.1) explique intuitivement la nécessité de cette propriété [62], la définition mathématique est donnée

en appendice.

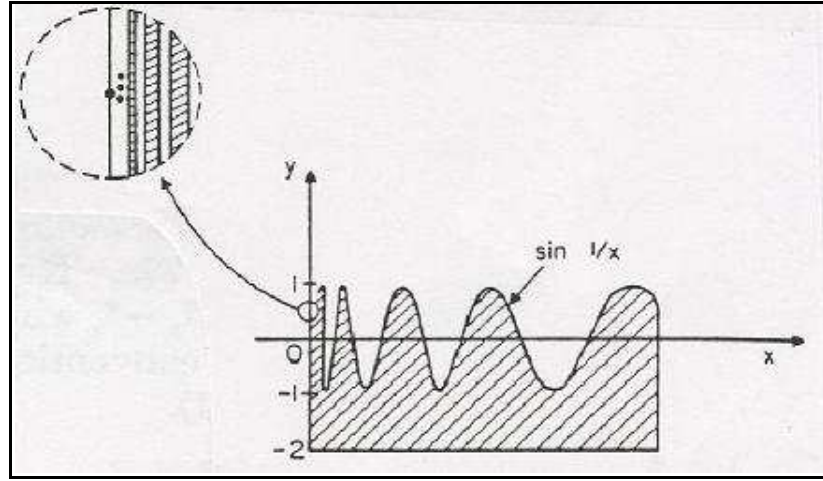


FIG. A.1: Ensemble non semi-analytique [62].

#### A.0.4 Régularité de l'opération

Comme les solides sont modélisés par des ensembles réguliers compacts, les opérations sur les solides doivent préserver cette propriété afin que le résultat soit encore un solide. Les opérations conventionnelles sur les ensembles ne vérifient pas ce critère (voir FIG. A.2). On considère alors l'intersection régularisée des objets définie comme précédemment par :

$$X \cap^* Y = fe(int(X \cap Y)).$$

L'intersection régularisée préserve la propriété de régularité sur l'ensemble résultat.

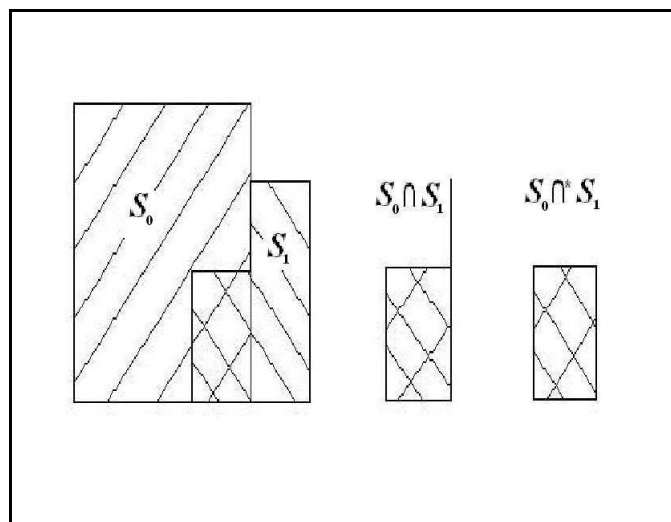


FIG. A.2: Intersection conventionnelle versus régularisée

# Annexe B

## Définitions liées à la topologie algébrique

Les définitions et les figures (FIG. B.1 et B.2), ont été prises dans [61].

1. Un *simplexe* (fermé) de dimension  $k$  (ou un  $k$ -simplexe) est un ensemble de points  $P$  dans  $E_n$  tels que

$$P = A_0.V_0 + A_1.V_1 + \dots + A_k.V_k,$$

où les scalaires  $A_i$  vérifient les deux conditions  $A_i \geq 0$  pour  $i = 0, \dots, k$  et  $\sum_0^k A_i = 1$ , et les  $k + 1$  points  $V_0, V_1, \dots, V_k$  sont tels que les vecteurs  $V_i - V_0$ ,  $i = 1, k$  sont linéairement indépendants. Les points sont appelés sommets du simplexe. En modélisation des solides, les simplexes utilisés sont de dimension inférieure ou égale à 3 (FIG. B.1).

2. Les *faces* d'un  $k$ -simplexe dont les sommets sont  $V_0, V_1, \dots, V_k$ , sont les simplexes définis par les sous-ensembles de l'ensemble  $\{V_0, V_1, \dots, V_k\}$ .

Par exemple, le triangle (FIG. B.1.c) a une  $2 - D$  face (le triangle lui-même), trois  $1 - D$  faces définis par les arêtes  $V_0V_1$ ,  $V_0V_2$  et  $V_1V_2$  et finalement trois  $0 - D$  faces qui sont les sommets eux-mêmes.

3. Un *complexe simplicial* (fini)  $K$  (FIG. B.2) est une ensemble de simplexes de  $E^n$  tels que :
  - si  $\mathbf{S}$  est un simplexe de  $K$  alors les faces de  $\mathbf{S}$  sont aussi des faces de  $K$ ,

- deux simplexes quelconques de  $K$  sont soit disjoints ou s'intersectent exactement sur une face commune.

La dimension d'un complexe simplicial  $K$  est la dimension maximale de ses simplexes.

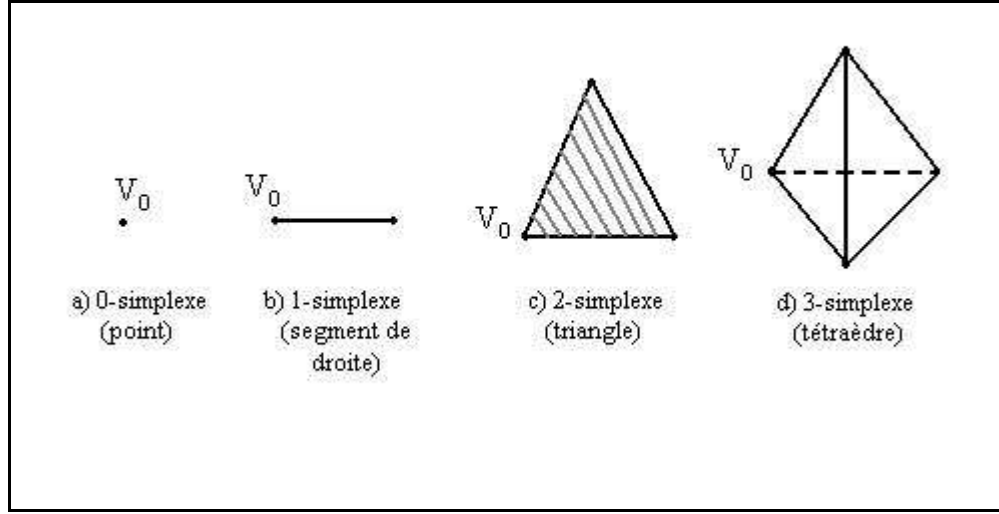


FIG. B.1: Simplexes utilisés

4. Un  $k$ -cycle (mod 2) sur un complexe  $K$  est la somme formelle

$$C = A_1.S_1 + \dots + A_k.S_k,$$

où les scalaires  $A_i$  sont les valeurs 0 ou 1 et les  $S_i$  sont les  $k$ -simplexes distincts de  $K$  dont la décomposition sous forme de somme formelle de simplexes de dimension  $k - 1 \bmod 2$  est nulle. Les sommes formelles (FIG. B.3)  $e_1 + e_2 + e_3$ ,  $e_3 + e_4 + e_5$  et  $e_5 + e_6 + e_7$ , sont des 1-cycles car on a, par exemple,

$$e_1 + e_2 + e_3 = V_1 + V_2 + V_2 + V_3 + V_3 + V_1 = 0 \pmod{2}.$$

De même,  $t_1 + t_2 + t_3 + t_4$  et  $t_1 + t_2 + t_3 + t_4 + t'_1 + t'_2 + t'_3 + t'_4$  sont deux 2-cycles car pour la première somme, on a, (mod(2))

$$t_1 + t_2 + t_3 + t_4 = (e_1 + e_2 + e_3) + (e_3 + e_4 + e_5) + (e_1 + e_5 + e_6) + (e_2 + e_4 + e_6) = 0$$

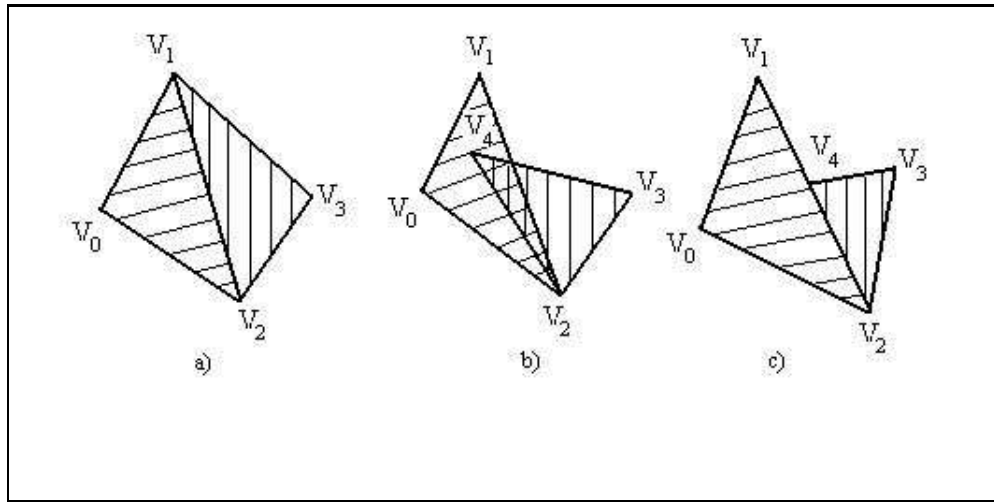


FIG. B.2: Collection de simplexes où  
 a) est un complexe simpliciale mais ni b), ni c) ne le sont

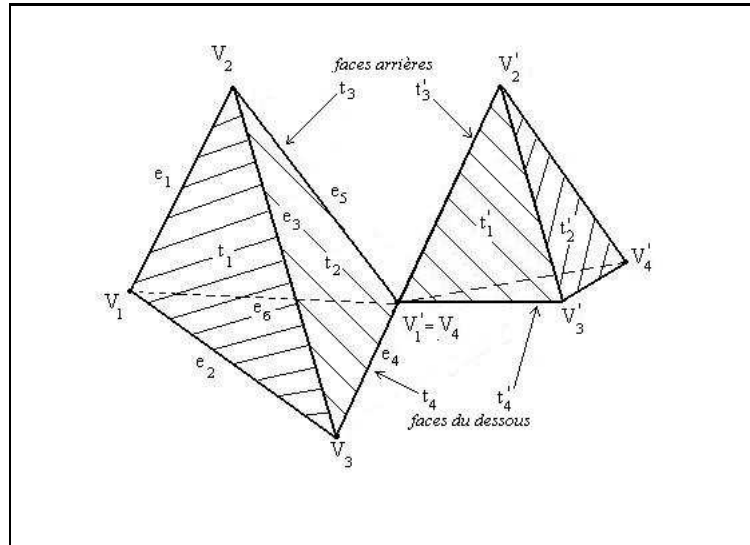


FIG. B.3:  $k$ -cycles sur un complexe simpliciale.



La notion de *complexe cellulaire* est une généralisation de la notion de *complexe simplicial* où les *cellules* remplacent les *simplexes*. Un *2-cell* est un ‘2-simplexe où le triangle est remplacé par un polygône mais un *1-cell* est un 1-simplexe (une arête) et *0-cell* est un 0-simplexe (un sommet).