

**Titre:** Champ distance d'un solide paramétrique  
Title:

**Auteur:** Xavier Mankovsky  
Author:

**Date:** 2009

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Mankovsky, X. (2009). Champ distance d'un solide paramétrique [Master's thesis, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/173/>

## Document en libre accès dans PolyPublie

Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/173/>  
PolyPublie URL:

**Directeurs de recherche:** François Guibault  
Advisors:

**Programme:** Génie informatique  
Program:

UNIVERSITÉ DE MONTRÉAL

CHAMP DISTANCE D'UN SOLIDE PARAMÉTRIQUE

XAVIER MANKOVSKY

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE INFORMATIQUE)

OCTOBRE 2009

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

CHAMP DISTANCE D'UN SOLIDE PARAMÉTRIQUE

présenté par: MANKOVSKY Xavier  
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées  
a été dûment accepté par le jury d'examen constitué de:

Mme CHERIET Farida, Ph.D., présidente

M. GUIBAULT François, Ph.D., membre et directeur de recherche

M. CAMARERO Ricardo, Ph.D., membre

À mes parents,

## REMERCIEMENTS

L'élaboration d'un projet de recherche est une expérience scientifique des plus enrichissantes. Mais c'est aussi une expérience humaine importante, particulière et inoubliable.

Ce mémoire n'aurait jamais pu être réalisé sans le soutien actif d'un entourage exceptionnel. Je tiens à exprimer ma profonde gratitude et mes sincères remerciements à toutes les personnes qui ont rendu ce travail possible.

A mon directeur de recherche, Monsieur Francois Guilbault, qui m'a guidé et m'a accordé sa confiance tout au long de mes travaux. Ses compétences scientifiques et ses qualités humaines, sa disponibilité et sa patience m'ont permis de mener à bien mon projet de recherche.

A Madame Farida Cherrier pour avoir accepté d'assurer la présidence du jury et pour avoir pris connaissance de mes travaux de recherche.

A Monsieur Ricardo Camarero pour ses conseils sur les méthodes numériques et pour avoir accepté d'être membre du jury.

Je remercie Christophe Tribes pour ses conseils en optimisation et Jean-François Dubé pour son aide sur les méthodes de projection.

Je veux également remercier tous mes professeurs de l'École qui m'ont donné une formation scientifique rigoureuse. J'en profite pour exprimer ma reconnaissance au personnel de la bibliothèque pour leur disponibilité.

Je remercie mes parents pour leur soutien et encouragements tout au long de ces longues années d'études supérieures. Sans eux, rien de tout cela n'aurait été envisageable.

Je tiens également à remercier mon frère, Jean-David, qui a toujours su être là

quand il le fallait.

Je remercie mes collègues du laboratoire, Yuanli Wang, Man Wang, Melisa Côté pour leur soutien et en particulier Anthony Haroutunian et Yvan Mokwinski, qui, en plus d'avoir été de bons collègues sont devenus de bons amis.

Je tiens aussi à remercier mes amis, Élaine, Simone, Matej, Manal, Patrick, Mathieu et Frédéric pour les nombreuses bonnes soirées que nous avons passées ensemble.

## RÉSUMÉ

Ce mémoire présente une méthode novatrice de calcul du champ distance d'un solide paramétrique. Bien que de nombreux travaux aient déjà été préalablement réalisés sur le champ distance d'une géométrie polygonale, aucune solution n'avait encore été apportée pour obtenir celui d'une géométrie paramétrique.

Le champ distance est un ensemble compact et fermé d'un espace euclidien, pour lequel est définie en tout point la distance euclidienne minimale permettant de rejoindre la surface d'un solide.

Le champ distance est calculé numériquement en résolvant une équation de transport basée sur l'équation Eikonal. L'équation Eikonal est une équation aux dérivées partielles du premier ordre, dites d'Hamilton-Jacobi. La résolution de l'équation Eikonal est un problème hyperbolique et non linéaire. Notre méthode repose sur le transport de conditions limites qui dépendent de la géométrie.

L'information se propage le long des courbes caractéristiques à partir des conditions limites. Le caractère non linéaire du problème conduit à la formation de chocs, selon la loi de conservation hyperbolique.

Nous avons utilisé une méthode de balayage, appelée *Fast-Sweeping*, pour résoudre l'équation Eikonal. Cette méthode consiste à discréteriser l'espace autour du solide à l'aide d'une grille cartésienne, puis à propager les conditions frontières en résolvant numériquement l'équation Eikonal selon un schéma de Godunov. La méthode de résolution peut être subdivisée en deux étapes. Dans un premier temps, il est nécessaire d'initialiser les conditions frontières. Dans un deuxième temps, l'équation Eikonal est résolue en utilisant plusieurs balayages alternés de Gauss-Seidel.

L'initialisation des conditions frontières est l'étape la plus importante et aussi la plus difficile à mettre en œuvre lors du calcul du champ distance d'un solide.

Les conditions limites sont constituées des nœuds limitrophes à la géométrie. Nous avons donc dû développer un algorithme permettant de détecter et d'initialiser précisément les nœuds frontières. Ces nœuds sont initialisés avec la valeur de la plus courte distance euclidienne permettant de rejoindre le solide. Cette distance minimale est obtenue en mesurant la longueur entre le nœud et sa projection sur la surface du solide.

Nous avons développé une méthode précise de projection sur une surface de Bézier.

Contrairement aux méthodes communément admises, qui calculent la projection à l'aide d'un critère d'orthogonalité, notre approche est basée sur la minimisation d'une distance euclidienne.

Dans le cas d'un solide fermé, l'orientation du champ distance est déterminée en propageant l'orientation des noeuds frontières. La localisation des noeuds intérieurs et des noeuds extérieurs est déterminée grâce de l'orientation géométrique de la surface.

Nous avons validé la précision du champ distance en effectuant une analyse de convergence de l'erreur. Nous avons montré que notre méthode est plus précise que les méthodes qui utilisent un maillage surfacique.

Nous avons validé la robustesse de l'orientation du champ distance sur plusieurs géométries où la localisation par rapport à la surface d'un point quelconque de l'espace est connue sans ambiguïté.

Contrairement aux travaux antérieurs, notre méthode permet de calculer le champ distance d'un solide directement à partir des surfaces paramétriques. Ainsi, l'erreur de représentation est très faible.

Les mesures effectuées ont montré que les temps de calcul des différents algorithmes développés sont linéaires par rapport à la discrétisation utilisée. Le temps de calcul a été fortement amélioré et permet de calculer des champs distances de grande taille.

Le champ distance à nombreuses applications en géométrie numérique. Sans vouloir remplacer les modèles paramétriques, nous croyons que le champ distance est une représentation complémentaire qui offre de nombreux avantages, notamment celui de représenter de façon implicite des géométries de topologie quelconque. Il permet aussi de modéliser les axes médians et de construire des surfaces parallèles.

## ABSTRACT

In this thesis we propose a novel method for computing the distance field of parametric shapes. Although many works related to distance field of polygonal geometries have been published, no solution has still been brought to obtain the distance field of parametric geometries.

Distance fields are defined as spatial fields of scalar distances to a surface geometry. At each point within the field, we know the distance from that point to the closest point on any object within the domain. In addition to distance, other properties may be derived from the distance field, such as the direction to the surface. When the distance field is signed, we may also determine if the point is internal or external to objects within the domain.

There is no closed form analytical formulas for computing the distance field of a manifold. Numerical methods are then necessary for computing the distance field of such complex shapes. We use a front propagation method using the Eikonal equation to propagate the parametric surfaces. We use an iterative algorithm, called the Fast Sweeping method, for computing the numerical solution for Eikonal equation on a rectangular grid. The main idea of this method is to use a nonlinear Godunov upwind difference scheme to discretize the partial differential equation, and Gauss-Seidel iterations. Information propagates along characteristics from the boundary. Due to the nonlinearity of the propagation problem, characteristics may intersect like the formation of shocks in a hyperbolic conservation law.

Boundary conditions are defined as the set of the closest grid nodes from the parametric surfaces. The initialisation of boundary conditions is the most difficult and the upmost critical step.

Indeed we present a novel method for detecting boundary nodes on a rectangular grid using grid properties and projection criterias. We also present a new algorithm for the projection of points on rational Bezier surfaces using an euclidien distance optimisation.

We implement a new post-processing algorithm to propagate the sign of boundary nodes to non-boundary nodes of the cartesien grid.

We validate our method with an error convergence analysis. We present numerical results to verify our analysis and show a few applications. We prove the convergence

and error estimates for the distance function. We also prove that computing the distance field of a parametric shape is more accurate than computing the same distance field from a polygonal geometry.

The robustness of the sign has been verified using geometric models for which point in solid test is well known.

## TABLE DES MATIÈRES

DÉDICACE . . . . .	iii
REMERCIEMENTS . . . . .	iv
RÉSUMÉ . . . . .	vi
ABSTRACT . . . . .	viii
TABLE DES MATIÈRES . . . . .	x
LISTE DES FIGURES . . . . .	xiv
LISTE DES NOTATIONS ET DES SYMBOLES . . . . .	xvi
LISTE DES TABLEAUX . . . . .	xvii
LISTE DES ANNEXES . . . . .	xviii
INTRODUCTION . . . . .	1
CHAPITRE 1 MODÉLISATION GÉOMÉTRIQUE DU SOLIDE . . . . .	9
1.1 Types de représentations géométriques . . . . .	9
1.2 Représentation paramétrique des surfaces gauches . . . . .	11
1.2.1 Surfaces B-splinaires . . . . .	11
1.2.1.1 Fonctions de bases B-splinaires . . . . .	12
1.2.1.2 Dérivées des fonctions de bases B-splinaires . . . . .	12
1.2.1.3 Formulation d'une surface B-splinaire . . . . .	12
1.2.2 Surfaces <i>NURBS</i> . . . . .	13
1.2.2.1 Formulation homogène des surfaces <i>NURBS</i> . . . . .	14
1.2.2.2 Dérivées des surfaces <i>NURBS</i> . . . . .	14
1.2.2.3 Évaluation numérique des surfaces <i>NURBS</i> . . . . .	15
1.2.2.4 Subdivision des surfaces <i>NURBS</i> . . . . .	15
1.2.3 Surfaces de Bézier rationnelles . . . . .	15
1.2.4 Orientation géométrique d'une surface paramétrique . . . . .	16

1.3	Modèle <i>B-Rep</i> . . . . .	17
1.4	Modèle de représentation implicite . . . . .	18
1.4.1	Définition d'une surface implicite . . . . .	18
1.4.2	Opérations de construction de solides . . . . .	20
1.4.3	Visualisation de surfaces implicites . . . . .	21
1.4.3.1	Le lancer de rayon . . . . .	21
1.4.3.2	Le rendu discret . . . . .	22
1.4.3.3	La facettisation . . . . .	22
1.4.4	Représentation paramétrique versus implicite . . . . .	23
1.4.5	Transformation vers une représentations implicite . . . . .	23
 CHAPITRE 2 CHAMP DISTANCE D'UN SOLIDE . . . . .		25
2.1	Définition du champ distance d'un solide . . . . .	25
2.1.1	Champ distance continu . . . . .	26
2.1.2	Champ distance discret . . . . .	27
2.2	Méthodes de calcul du champ distance discret . . . . .	28
2.3	Équation Eikonal . . . . .	30
2.3.1	Formulation Eikonal de la fonction distance . . . . .	31
2.4	Intersection des caractéristiques . . . . .	32
2.5	Résolution de l'équation Eikonal . . . . .	33
2.5.1	Méthodes de résolution par avance de front . . . . .	34
2.5.1.1	Ensemble de niveaux ou <i>Level Set</i> . . . . .	35
2.5.1.2	<i>Fast Marching</i> . . . . .	36
2.5.2	Résolution par éléments finis . . . . .	37
2.5.3	Méthodes de résolution par balayage . . . . .	38
2.5.3.1	<i>Fast Sweeping</i> . . . . .	38
2.5.4	Schéma de Lax-Friedrichs . . . . .	40
2.5.5	Schéma de Godunov . . . . .	43
2.5.5.1	Description de la méthode du <i>Fast Sweeping</i> . . . . .	43
2.5.5.1.1	Algorithme du <i>Fast Sweeping</i> . . . . .	46
2.5.5.2	Convergence du <i>Fast Sweeping</i> . . . . .	46
2.5.5.3	Propriétés du <i>Fast Sweeping</i> . . . . .	47
2.5.5.4	<i>Fast Sweeping</i> d'ordre supérieur . . . . .	48
2.5.6	Choix de la méthode retenue . . . . .	48

2.6	Propagation d'erreur à partir des surfaces discrétisés . . . . .	49
2.6.1	Erreur de représentation des surfaces discrétisées . . . . .	50
2.6.2	Sensibilité aux conditions frontières . . . . .	50
2.7	Résumé . . . . .	51
CHAPITRE 3 PROJECTION ORTHOGONALE . . . . .		53
3.1	Projection orthogonale . . . . .	53
3.2	Projection d'un point sur une surface paramétrique . . . . .	56
3.2.1	Projection sur une surface . . . . .	57
3.2.2	Revue des méthodes de projection avancées . . . . .	59
3.2.3	Problèmes de robustesse existants . . . . .	61
3.2.3.1	Normales dégénérées . . . . .	62
3.3	Projection par optimisation d'une distance . . . . .	63
3.3.1	Expression de la fonction donnant la projection . . . . .	63
3.3.2	Minimisation de la distance . . . . .	66
3.3.3	Choix du point de départ . . . . .	67
3.3.4	Solutions multiples au problème de la projection . . . . .	67
3.3.5	Traitements des solutions multiples . . . . .	68
3.3.5.1	Relations entre les critères de convergence . . . . .	69
3.4	Résumé . . . . .	69
CHAPITRE 4 CALCUL DU CHAMP DISTANCE D'UN SOLIDE PARAMÉTRIQUE . . . . .		71
4.1	Rappel des objectifs . . . . .	71
4.2	Méthodologie . . . . .	72
4.3	Subdivision de la surface du solide . . . . .	72
4.4	Définition et discréétisation du domaine de calcul . . . . .	72
4.5	Initialisation des conditions frontières . . . . .	74
4.5.1	Détection des nœuds frontières . . . . .	74
4.5.1.1	Évaluer la profondeur de tessellation . . . . .	79
4.5.2	Initialisation des conditions frontières . . . . .	81
4.5.2.1	Point de départ pour la projection . . . . .	82
4.5.2.2	Projection des nœuds frontières . . . . .	83
4.5.3	Orientation géométrique des nœuds frontières . . . . .	84

4.5.4	Algorithme d'initialisation des conditions frontières . . . . .	84
4.6	Calcul du champ distance . . . . .	84
4.7	Orientation du champ distance . . . . .	86
4.8	Résumé . . . . .	87
CHAPITRE 5 ANALYSE ET VALIDATION DES RÉSULTATS . . . . .		89
5.1	Mesure de l'erreur . . . . .	89
5.2	Validation du <i>Fast-Sweeping</i> . . . . .	90
5.3	Modèle bicubique d'une sphère . . . . .	91
5.4	Validation de champ distance d'un solide . . . . .	92
5.4.1	Validation des conditions frontières . . . . .	92
5.4.1.1	Validation de la détection des nœuds frontières . .	93
5.4.1.2	Validation de l'initialisation des nœuds frontières .	93
5.4.1.3	Validation de l'orientation des conditions frontières	94
5.4.2	Analyse de convergence de l'erreur du champ distance . .	94
5.4.3	Validation de l'orientation du champ distance . . . . .	95
5.5	Exemples de champ distance . . . . .	96
5.5.1	Cylindre . . . . .	96
5.5.2	Solide de révolution d'une parabole . . . . .	98
5.5.3	Le <i>Utah Teapot</i> . . . . .	102
5.6	Performance et temps de calcul . . . . .	104
5.7	Résumé . . . . .	104
CONCLUSION . . . . .		108
BIBLIOGRAPHIE . . . . .		115
ANNEXES . . . . .		120

## LISTE DES FIGURES

Figure 1	Axe et surface médiane. . . . .	2
Figure 2	Opération morphologique . . . . .	3
Figure 3	Maillage obtenu à l'aide de <i>DistMesh</i> . . . . .	3
Figure 4	Erreur de représentation due à la discréétisation. . . . .	5
Figure 5	Zones d'expansion . . . . .	5
Figure 1.1	Classification des méthodes de représentation géométrique. .	10
Figure 1.2	Orientation pour une représentation paramétrique. . . . .	16
Figure 1.3	Orientation d'une surface. . . . .	16
Figure 1.4	Orientation des boucles pour former un trou topologique. .	18
Figure 1.5	Opérations de construction de solide. . . . .	20
Figure 1.6	Lancer de rayon. . . . .	22
Figure 1.7	Les quatorze cas de base du <i>Marching Cubes</i> . . . . .	23
Figure 2.1	Orientation du champ distance d'un parcours simple fermé.	26
Figure 2.2	Maillage pour le champ distance. . . . .	28
Figure 2.3	Maillage surfacique d'une boîte de vitesse. . . . .	29
Figure 2.4	Zone d'expansion et zone de choc. . . . .	33
Figure 2.5	Classification des méthodes de résolution de l'équation Eikonal.	34
Figure 2.6	Directions du balayage en deux dimensions. . . . .	39
Figure 2.7	Caractéristiques de l'équation Eikonal. . . . .	40
Figure 2.8	Représentation des noeuds de la grille cartésienne formant les conditions frontières. . . . .	44
Figure 2.9	Erreur commise lors de la projection d'un point sur une géométrie discréétisée. . . . .	50
Figure 2.10	Représentation de la propagation de l'erreur dans un cas unidimensionnel. . . . .	51
Figure 3.1	Convexité d'un ensemble. . . . .	54
Figure 3.2	Type de solution pour la projection. . . . .	55
Figure 3.3	Projection orthogonale sur une courbe paramétrique. . . . .	56
Figure 3.4	Étapes pour la projection. . . . .	60
Figure 3.5	Axe médian d'un rectangle. . . . .	67
Figure 3.6	Solution multiples lors de la projection. . . . .	68

Figure 3.7	Domaine paramétrique contenant deux minima. . . . .	69
Figure 4.1	Expansion du domaine de calcul. . . . .	73
Figure 4.2	Détection des nœuds frontières sur un artefact aigu. . . . .	75
Figure 4.3	Nœuds frontières d'un cercle. . . . .	76
Figure 4.4	Conditions frontières pour un segment. . . . .	78
Figure 4.5	Représentation de l'erreur maximale $\epsilon$ . . . . .	79
Figure 4.6	Triangle paramétrique. . . . .	82
Figure 4.7	Point de départ pour la projection. . . . .	83
Figure 4.8	Orientation géométrique d'un nœud frontière. . . . .	84
Figure 4.9	Initialisation des conditions frontières. . . . .	85
Figure 4.10	Propagation des conditions frontières. . . . .	86
Figure 4.11	Orientation du champ distance. . . . .	87
Figure 5.1	Convergence de l'erreur pour la méthode du <i>Fast-Sweeping</i> . . . . .	91
Figure 5.2	Polygones de contrôle d'une sphère. . . . .	92
Figure 5.3	Nœuds frontières d'une sphère. . . . .	93
Figure 5.4	Erreur du champ distance. . . . .	95
Figure 5.5	Champ distance d'une sphère. . . . .	96
Figure 5.6	Comparaison entre des isosurfaces. . . . .	97
Figure 5.7	Nœuds frontières d'un cylindre. . . . .	98
Figure 5.8	Erreur du champ distance du cylindre et de la surface de révolution. . . . .	99
Figure 5.9	Géométrie du solide de révolution. . . . .	100
Figure 5.10	Nœuds frontières d'un solide de révolution. . . . .	100
Figure 5.11	Surfaces iso du champ distance d'une surface de révolution. . . . .	101
Figure 5.12	Champ distance du <i>TeaPot</i> . . . . .	102
Figure 5.13	Surfaces iso du champ distance du <i>TeaPot</i> . . . . .	103
Figure 5.14	Temps de calcul. . . . .	106
Figure I.1	Illustration du triangle $T(s, t)$ . . . . .	121
Figure I.2	Division du plan $st$ . . . . .	122
Figure I.3	Courbes de niveaux de la fonction $Q$ pour la région 1. . . . .	123
Figure I.4	Courbe de niveau possible pour la région 2. . . . .	125

## LISTE DES NOTATIONS ET DES SYMBOLES

### Notations relatives au champ distance

- $\phi$  : distance euclidienne.
- $\Gamma$  : frontière d'une géométrie.
- $\nabla$  : gradiant de  $\phi(\vec{x})$ .
- $|\cdot|$  : norme euclidienne.

### Notations relatives aux surfaces paramétriques

- $S$  : surface paramétrique.
- $u$  : paramètre d'une surface.
- $v$  : paramètre d'une surface.
- $P_{i,j}$  : nœud d'un polygone de contrôle.
- $\lceil \cdot \rceil$  : valeur entière supérieure.

### Notations relatives à l'analyse d'erreur

- $\|e\|_\infty$  : erreur maximale commise.
- $\|e\|_{l^2}$  : norme discrète  $l^2$  de l'erreur.
- $\|e\|_{l^1}$  : norme discrète  $l^1$  de l'erreur.

**LISTE DES TABLEAUX**

Tableau 1.1	Comparaison des représentations implicites et paramétriques.	24
Tableau 5.1	Convergence de l'erreur pour la méthode du Fast-Sweeping.	90
Tableau 5.2	Analyse de l'erreur des nœuds frontières. . . . .	94
Tableau 5.3	Analyse de convergence de l'erreur . . . . .	94
Tableau 5.4	Analyse de l'erreur pour le cylindre. . . . .	97
Tableau 5.5	Analyse de l'erreur pour la roue. . . . .	99
Tableau 5.6	Performance et temps de calcul. . . . .	105

**LISTE DES ANNEXES**

ANNEXE I	PROJECTION SUR UNE FACETTE TRIANGULAIRE	120
I.1	Distance minimale d'un point à un triangle	120
I.2	Distance calculée analytiquement	120
ANNEXE II	SURFACES <i>NURBS</i>	127
II.1	Propriétés des surfaces <i>NURBS</i>	127
II.2	Dérivées de surfaces <i>NURBS</i>	128
II.3	Évaluation numérique des surfaces <i>NURBS</i>	130
II.4	Insertion de nœuds dans une surface <i>NURBS</i>	131
ANNEXE III	SURFACES DE BÉZIER RATIONNELLES	135
III.1	Polynômes de Bernstein	135
III.1.1	Formulation des surfaces de Bézier	136
III.2	Subdivision matricielle des surfaces de Bézier	137

## INTRODUCTION

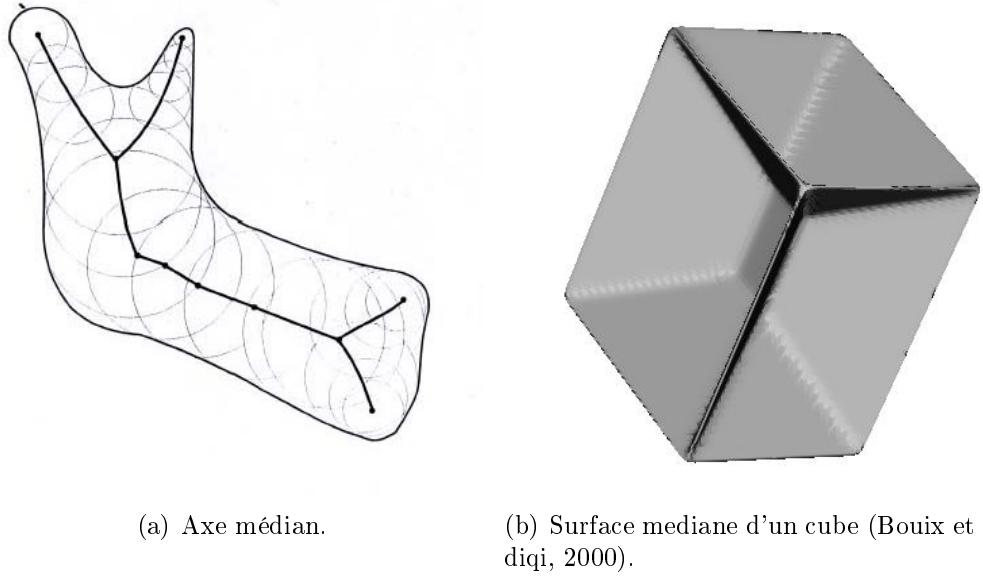
### Contexte du projet

Les méthodes de représentation surfaciques d'un solide sur ordinateur sont très variées. Cependant, dans certains contextes, il est nécessaire de représenter le volume entourant la surface d'un solide. Le *champ distance* est une représentation implicite permettant de modéliser la géométrie, l'espace extérieur ainsi que le volume intérieur d'un solide de topologie quelconque, en donnant localement une indication sur la proximité du solide.

On définit le champ distance d'un solide comme un ensemble compact et fermé d'un espace euclidien, contenant un solide, pour lequel est définie en tout point la distance euclidienne minimale, ainsi que la direction, permettant de rejoindre la surface du solide. Le champ distance est orienté si le solide est fermé, ce qui permet de connaître la localisation d'un point de l'espace par rapport au solide. La distance est positive à l'extérieur du solide et négative à l'intérieur.

La *carte de distance* d'un solide possède de nombreuses applications en géométrie numérique et en ingénierie. Par exemple, les propriétés du champ distance permettent de déterminer efficacement les axes médians ou les surfaces médianes d'un solide. L'axe médian, appelé aussi *squelette de distance*, est une méthode permettant de représenter la géométrie d'un solide en déterminant son squelette topologique. En deux dimensions, l'axe médian d'une courbe fermée est le lieu des centres des cercles qui sont tangents à la courbe en au moins deux points. L'axe médian peut être généralisé en  $n$  dimensions à l'aide des hypersurfaces. Les cercles en deux dimensions et les sphères en trois dimensions sont alors remplacés par des hypersphères de dimension  $n$ . Dans un champ distance, le locus de l'axe médian coïncide avec l'intersection des caractéristiques. Ainsi, nous pouvons, une fois la carte de distance d'un solide calculée, localiser les axes médians de la géométrie.

Le champ distance permet d'appliquer des opérations morphologiques d'érosion, de dilatation, d'ouverture et de fermeture à la géométrie d'un solide (Jones *et al.*, 2006). Une opération d'érosion permet de supprimer une partie extérieure du so-

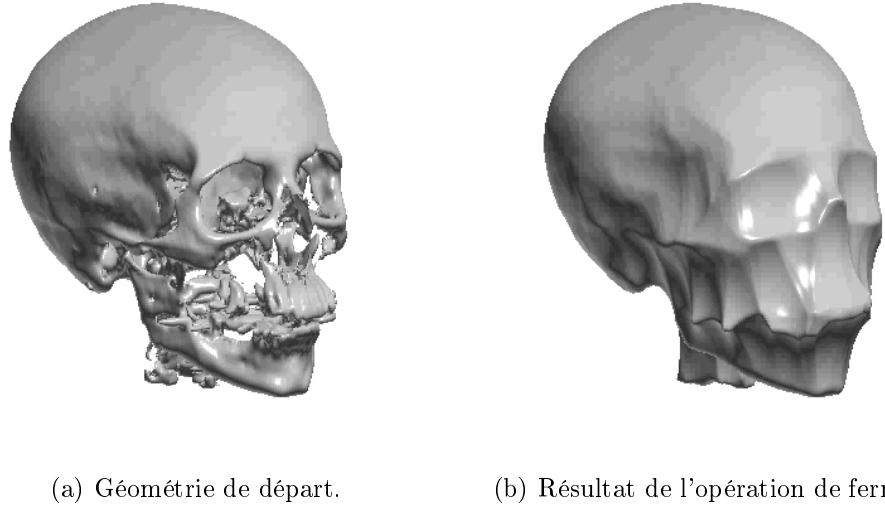


**Figure 1:** Axe et surface médiane.

lide, alors qu'une dilatation ajoute une nouvelle extension à la géométrie du solide. Une opération d'ouverture permet d'élargir les entailles et les cavités de la géométrie. L'opération d'ouverture est une érosion suivie d'une dilatation. L'opération de fermeture, illustrée à la figure 2(b), va, quant à elle, refermer les cavités et adoucir les artefacts aigus. Cette opération est formée d'une dilatation suivie d'une érosion. Les opérations morphologiques sont plus faciles à réaliser sur un modèle implicite, comme le champ distance, que sur un modèle paramétrique. En effet, la continuité lors de l'assemblage de plusieurs surfaces paramétriques est difficile à garantir.

De plus, le champ distance peut représenter des solides de topologie quelconques, contrairement aux surfaces paramétriques où la représentation d'un solide muni de trous nécessite l'utilisation de multiples surfaces dont l'assemblage doit être régi par un modèle topologique cohérent. Dans le cas du champ distance, la topologie du solide et la géométrie sont imbriquées.

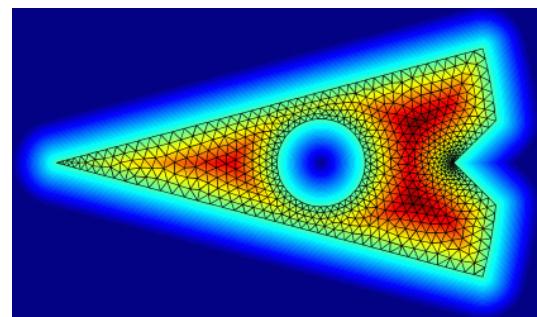
Une autre application possible au champ distance est la conception de solide à l'aide de la modélisation volumique. La géométrie du solide est le résultat d'une suite d'opérations booléennes sur le champ distance de formes géométriques élémentaires.



(a) Géométrie de départ. (b) Résultat de l'opération de fermeture.

**Figure 2:** Opération morphologique de fermeture sur un modèle de crâne (Jones *et al.*, 2006).

Le champ distance est une représentation géométrique permettant de développer de nouvelles méthodes de décomposition de domaine et de génération de maillage. Persson (2004) a développé une méthode de génération de maillage non struture pour les géométries implicites en utilisant un champ distance orienté. Un maillage initial est construit à l'aide de la méthode de Delaunay, puis le maillage est amélioré en résolvant itérativement une équation d'équilibre sur les arêtes des éléments (voir figure 3). Une métrique basée sur la courbure et le champ distance de la géométrie est calculé avant la génération de maillage et permet d'obtenir des éléments de taille optimale.



**Figure 3:** Maillage obtenu à l'aide de *DistMesh*.

Wang *et al.* (2005) a développé une méthode de génération de maillage par construction de peaux successives. Les maillages en peau sont utilisés en dynamique des fluides pour les problèmes d'écoulement. Le champ distance permet la construction de surfaces dites *parallèles*. Ainsi, on peut construire une frontière paramétrique située à une distance constante de la surface d'origine. Cette nouvelle frontière permet alors d'utiliser des méthodes de génération de maillage algébrique ou elliptique (Frey et George, 1999). En effet, le calcul des transformations transfinies nécessite un domaine fermé par quatre frontières paramétriques.

Le champ distance, de par ses propriétés, permet de concevoir tout une gamme d'outils mathématiques permettant de se déplacer dans le voisinage du solide représenté.

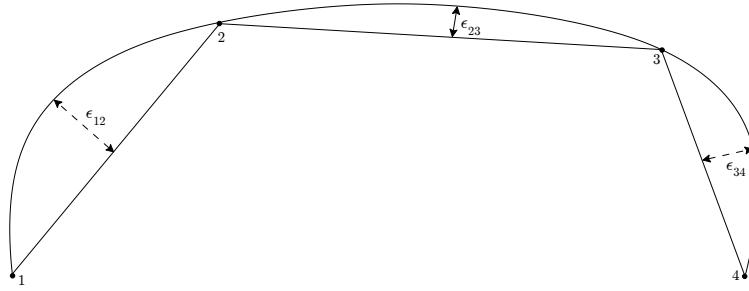
## Problématique

Nous cherchons à obtenir le champ distance d'un solide de topologie quelconque dont la surface est représentée par une géométrie paramétrique. Nos travaux se concentreront sur les géométries représentées à l'aide de surfaces B-splinaires rationnelles et non uniformes, communément appelées surfaces *NURBS* (*Non-Uniform Rational B-Splines*).

De nombreuses méthodes de calcul permettant d'obtenir le champ distance d'un solide (Jones *et al.*, 2006). Cependant, les méthodes développées jusqu'à ce jour reposent sur une représentation polygonale de la géométrie du solide. Or la discréétisation d'une surface paramétrique donne toujours lieu à une erreur de représentation. Comme nous le démontrerons à la section 2.6, cette erreur est intégralement propagée à travers tout le champ distance. La figure 4 représente une illustration de l'erreur maximale commise lors de la discréétisation d'une courbe.

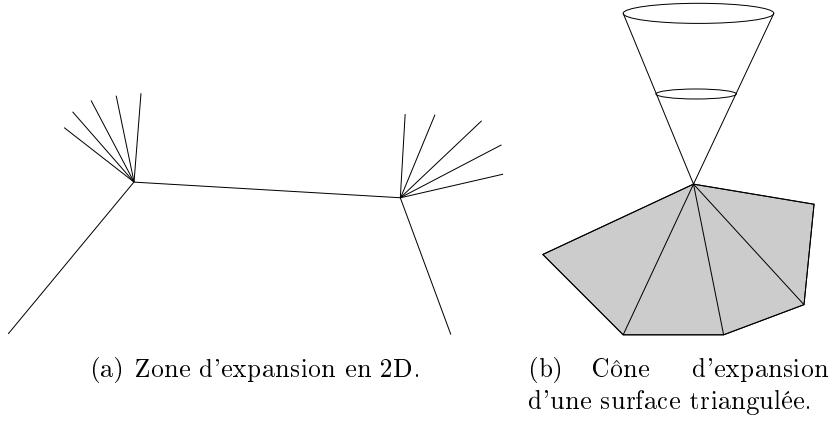
Aucune méthode n'a encore été proposée afin de calculer le champ distance d'un solide directement à partir d'une représentation paramétrique sans devoir préalablement effectuer une discréétisation polygonale de la surface.

Un autre inconvénient au calcul de la carte de distance à partir d'un maillage sur-



**Figure 4:** Erreur de représentation due à la discrétisation. La courbe est discrétisée en quatre points reliés par des segments. L'erreur maximale  $\epsilon_{ij}$  sur chaque intervalle entre le nœud  $i$  et le nœud suivant  $j$  est représentée.

facique du solide est la formation de zones d'expansion. Lors du calcul du champ distance, une géométrie polygonale génère de nombreuses zones d'expansion appelées aussi zones de raréfaction, au niveau des arêtes et des sommets, selon la loi de conservation hyperbolique. Dans une zone de raréfaction, on ne connaît qu'une solution faible et l'erreur est plus importante.



**Figure 5:** Représentation des zones d'expansion dans la solution due à la discrétisation. Dans les zones d'expansion, l'erreur est plus importante et seule une solution faible est connue. Une surface triangulée de continuité  $C^0$  génère des zones de raréfaction de forme conique.

Un autre problème est l'orientation du champ distance. Dans les travaux antérieurs, la surface triangulée n'était pas orientée. Pour orienter un nœud de la grille, il était nécessaire d'utiliser une méthode de lancer de rayon. Cette méthode nécessite de calculer l'intersection entre un rayon (une demi-droite) et le maillage surfacique. Les méthodes d'intersection sont difficiles à développer, principalement à cause des

problèmes de robustesse liés à la représentation des nombres et aux cas dégénérés (Segura et Feito, 1998).

L'objectif de ce travail de recherche est de développer une méthode de calcul permettant de minimiser l'erreur de représentation due à la discréétisation, en calculant le champ distance d'un solide directement à partir des surfaces paramétriques.

### **Hypothèses de recherche**

La problématique ayant été identifiée, il convient maintenant d'établir les hypothèses de recherche. Trois hypothèses de recherche vont guider ce projet.

Notre première hypothèse porte sur la représentation du solide. Nous utiliserons des solides représentés par des surfaces paramétriques. La représentation de la surface est donc exacte. L'utilisation de surfaces paramétriques permet d'initialiser les conditions frontières avec une meilleure précision. Le calcul du champ distance à partir de surfaces paramétriques devrait ainsi permettre d'améliorer la précision du champ distance.

Notre seconde hypothèse considère qu'il est possible de déterminer des conditions frontières proches de la surface permettant de propager la distance dans l'espace. Les conditions limites sont déterminées à l'aide d'une projection sur le solide. La capacité de propager l'information le long de caractéristiques à partir de conditions frontières est une propriété des équations hyperboliques de type Hamilton-Jacobi. L'existence et l'unicité de la solution de ces équations ont été démontrées dans Crandall et Lions (1983).

La troisième hypothèse consiste à vérifier que l'utilisation de surfaces paramétriques permet de rendre l'orientation du champ distance plus robuste.

### **Objectifs du projet**

Nous voulons que le champ distance préserve la continuité de la représentation paramétrique du solide. La méthode de calcul du champ distance devra permettre

un contrôle précis de l'erreur. Nous souhaitons développer une méthode de calcul du champ distance d'un solide qui n'introduit pas de discontinuités dans la géométrie continue originale.

Au niveau de l'implémentation, nous désirons obtenir une méthode rapide, dont le temps de calcul est linéaire par rapport à la dimension du problème.

Une représentation complète de la topologie d'un solide permet de distinguer l'intérieur de l'extérieur. Dans le cas d'une représentation paramétrique, l'orientation de la géométrie est déterminée par le modèle *B-Rep*. Pour une représentation implicite, c'est la valeur de la fonction potentielle qui permet la localisation d'un point de l'espace. Afin que notre représentation implicite d'un solide soit complète, il sera nécessaire d'orienter le champ distance.

Nous démontrerons la qualité du champ distance obtenu à l'aide de solides dont la géométrie, connue sous une forme analytique, permet de calculer facilement un champ distance exact. Une analyse de convergence de l'erreur montrera que notre méthode permet d'obtenir le champ distance d'un solide avec la résolution souhaitée.

## Présentation du mémoire

Le présent mémoire est divisé en cinq chapitres. Dans le premier chapitre, une revue des concepts liés à la modélisation géométrique du solide est présentée. La représentation paramétrique des surfaces gauches sera présentée, en particulier les surfaces *NURBS*. Les algorithmes d'évaluation, d'insertion de noeuds et de subdivision seront exposés. Ce chapitre traite aussi des méthodes de modélisation implicite du solide.

Le second chapitre passe en revue les méthodes de calcul du champ distance existantes. Les méthodes de calcul numérique les plus courantes pour calculer un champ distance seront présentées. La résolution de l'équation Eikonal sera particulièrement détaillée.

Au troisième chapitre, présente dans un premier temps les concepts mathématiques

de base de la projection orthogonale et, dans un deuxième temps, une revue des méthodes de projection sur les surfaces paramétriques. Enfin une nouvelle méthode de projection sur les surfaces *NURBS* sera proposée.

Le quatrième chapitre est consacré aux algorithmes développés pour calculer le champ distance d'une surface paramétrique. Les étapes permettant de calculer le champ distance d'un solide représenté par une surface paramétrique seront expliquées en détail.

Dans le cinquième chapitre rend compte des résultats obtenus. Une analyse de convergence permettra de valider les différentes méthodes numériques développées dans ce projet. Une analyse de performance des algorithmes de mise en oeuvre sera aussi présentée.

Enfin, certaines perspectives de travaux visant à améliorer la précision et l'efficacité de la méthodologie seront proposées dans la conclusion.

## CHAPITRE 1

### MODÉLISATION GÉOMÉTRIQUE DU SOLIDE

Ce chapitre vise à définir les concepts mathématiques liés à la représentation géométrique d'un solide. Nous présenterons dans un premier temps les différents outils de modélisation géométrique, en particulier de représentation paramétrique des surfaces gauches. Dans un deuxième temps, nous expliquerons comment ces surfaces peuvent aussi être représentées sous une forme implicite.

#### 1.1 Types de représentations géométriques

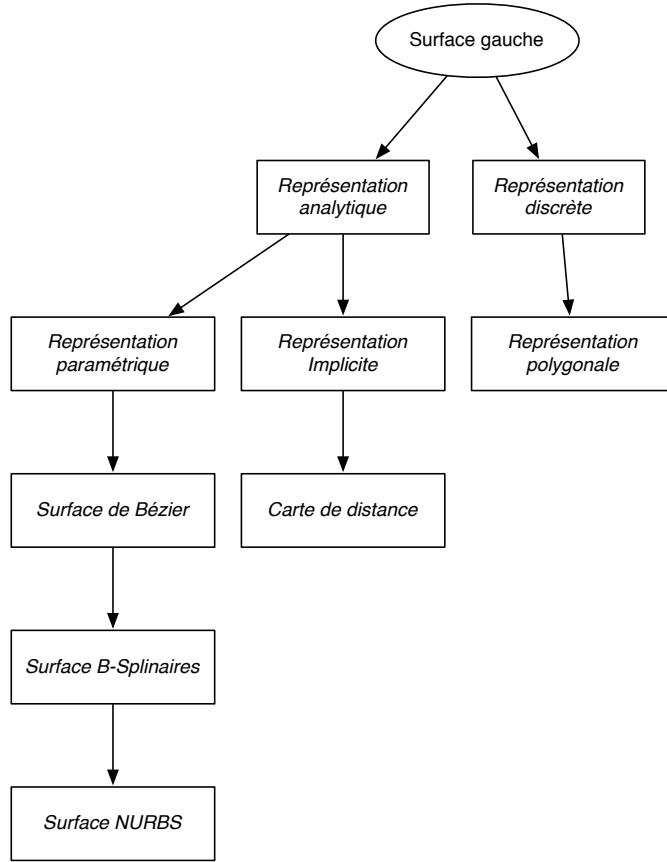
Les formes géométriques sont classifiées selon deux types. Le premier type est un assemblage de formes géométriques élémentaires telles que des plans ou des cylindres. Le deuxième type, regroupe quant à lui, les surfaces dites *gauches*. Une surface est gauche si elle ne peut pas être décrite par un système d'équations linéaires. Ces surfaces sont couramment utilisées dans l'industrie lors du processus de conception géométrique. Elles offrent aussi toute la flexibilité et le contrôle nécessaire à l'optimisation assistée par ordinateur.

On distingue principalement trois méthodes de représentation de surfaces gauches :

- Les représentations discrètes ;
- Les représentations paramétriques ;
- Les représentations implicites.

La figure 1.1 montre une hiérarchie des méthodes de représentation géométrique des surfaces gauches. Ces méthodes possèdent chacune leurs avantages et leurs inconvénients. Il est impossible de déterminer un modèle de représentation idéal. Il convient de choisir le bon mode de représentation en fonction des besoins et des contraintes liées au design et à la conception.

Les principales différences entre chaque méthode de représentation sont présentées ci-dessous :



**Figure 1.1:** Classification non exhaustive des méthodes de représentation géométrique des surfaces gauches.

- Une représentation implicite est fortement liée à un repère. Il est difficile de le changer, contrairement à une représentation paramétrique où l'on peut effectuer un changement de repère en conservant parfaitement la topologie, par exemple lors de transformations géométriques (translation, rotation).
- Une représentation paramétrique est plus facile à visualiser qu'une représentation implicite. L'affichage d'une géométrie paramétrique est réalisé en discrétilisant l'intervalle de paramétrisation, puis en évaluant l'équation paramétrique. En revanche, l'affichage d'une représentation implicite nécessite plus de calcul, car il faut résoudre une équation de niveau (voir section 1.4.3).
- La localisation spatiale des points est triviale dans une représentation implicite.

Il suffit d'évaluer la fonction implicite au point voulu et de déterminer la localisation spatiale à l'aide du signe de la valeur résultante. Une valeur négative indique que le point est à l'intérieur du solide. Dans le cas des représentations paramétriques, il est nécessaire d'utiliser un modèle géométrique complet et orienté comme un *B-Rep*, puis d'effectuer un calcul de la normale (voir section 1.3).

- La représentation des intersections et la construction d'une surface parallèle à une autre (*offsets*) se calculent plus facilement à l'aide d'une représentation implicite plutôt que paramétrique.

On trouve un résumé des avantages et des inconvénients pour chacune des méthodes à la figure 1.1. La représentation la plus couramment utilisée dans l'industrie pour représenter des surfaces gauches est la représentation paramétrique.

## 1.2 Représentation paramétrique des surfaces gauches

Les surfaces gauches sont facilement représentables à l'aide d'une fonction paramétrique. On distingue plusieurs méthodes de représentation paramétrique. Nous présenterons d'abord les surfaces B-splinaires, puis les surfaces *NURBS* (*Non-Uniform Rational Basis Splines*) et, enfin le cas particulier des surfaces de Bézier.

### 1.2.1 Surfaces B-splinaires

Une *spline* est une fonction polynomiale définie par morceaux. Les *B-splines* (*basis spline*) sont la généralisation des courbes de Bézier, elles sont à leur tour généralisées par les *NURBS*. Une B-spline est une combinaison linéaire de splines non-négatives à support compact minimal. Les surfaces B-splines forment un ensemble de méthodes de représentation paramétrique de surfaces.

### 1.2.1.1 Fonctions de bases B-splinaires

Soit  $U = \{u_0, \dots, u_m\}$  une séquence monotone non décroissante de nombres réels, tel que  $u_i \leq u_{i+1}$  pour  $i = 0, \dots, m - 1$ . Les valeurs  $u_i$  sont appelées des noeuds et  $U$  est un vecteur nodal.

On définit la  $i^e$  fonction de base B-splinaire de degré  $p$  (d'ordre  $p+1$ ), notée  $N_{i,p}(u)$  par :

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{si } u_i \leq u < u_{i+1}, \\ 0 & \text{autrement,} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \end{aligned}$$

Les fonctions d'interpolation (*blending functions*) B-splinaires possèdent plusieurs propriétés qui les rendent adéquates pour la modélisation mathématique des surfaces gauches (Piegl et Tiller, 1995).

### 1.2.1.2 Dérivées des fonctions de bases B-splinaires

La dérivée d'une fonction de base B-spline est donnée par :

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (1.1)$$

### 1.2.1.3 Formulation d'une surface B-splinaire

Une surface B-Spline est obtenue à l'aide d'une grille bidirectionnelle de points de contrôle, appelée *polygone de contrôle*. La surface B-spline est calculée en multipliant deux ensembles de fonctions de base B-splinaires :

$$\vec{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \vec{P}_{i,j}.$$

Chaque ensemble de fonctions de base est défini sur un vecteur nodal qui lui est propre :

$$\begin{aligned} U &= \left\{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{ni-p+1}, \underbrace{1, \dots, 1}_{p+1} \right\}, \\ V &= \left\{ \underbrace{0, \dots, 0}_{q+1}, u_{q+1}, \dots, u_{nj-q+1}, \underbrace{1, \dots, 1}_{q+1} \right\}. \end{aligned}$$

### 1.2.2 Surfaces *NURBS*

Les surfaces *NURBS* sont une généralisation des surfaces B-splines. Cependant les *NURBS* sont des fonctions rationnelles par morceaux plutôt que des fonctions polynomiales par morceaux comme les B-splines. On peut ainsi utiliser les *NURBS* pour représenter exactement des formes géométriques analytiques telles que les coniques. Les propriétés des surfaces *NURBS* sont données en annexe, à la section II.1.

Une surface *NURBS* de degré  $p$  dans la direction  $u$  et de degré  $q$  dans la direction  $v$  est une fonction vectorielle bivariée et rationnelle par morceau de la forme :

$$\vec{S}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} \vec{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}},$$

avec les bornes  $0 \leq u, v \leq 1$  sur la paramétrisation.

Les points  $\vec{P}_{i,j}$  forment un polygone de contrôle, ils sont associées les pondérations  $w_{i,j}$ . Les fonctions  $N_{i,p}(u)$  et  $N_{j,q}(v)$  sont les fonctions de base B-splinaires non rationnelles répectivement sur le vecteur nodal :

$$\begin{aligned} U &= \left\{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{ni-p+1}, \underbrace{1, \dots, 1}_{p+1} \right\}, \\ V &= \left\{ \underbrace{0, \dots, 0}_{q+1}, u_{q+1}, \dots, u_{nj-q+1}, \underbrace{1, \dots, 1}_{q+1} \right\}. \end{aligned}$$

On peut aussi définir la fonction de base rationnelle suivante :

$$R_{i,j}(u, v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u)N_{l,q}(v)w_{k,l}}.$$

Ainsi, on peut écrire la surface *NURBS* comme :

$$\vec{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) \vec{P}_{i,j}.$$

### 1.2.2.1 Formulation homogène des surfaces *NURBS*

Afin de calculer une surface *NURBS* dans l'espace projectif, on peut définir une nouvelle formulation à l'aide des coordonnées homogènes :

$$\vec{S}^w(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(u) \vec{P}_{i,j}^w,$$

où

$$\vec{P}_{i,j}^w = \begin{pmatrix} w_{i,j} \vec{P}_{i,j} \\ w_{i,j} \end{pmatrix}.$$

### 1.2.2.2 Dérivées des surfaces *NURBS*

Le calcul des dérivées des surfaces *NURBS* est très important. En effet, la dérivée d'une surface intervient dans la plupart des méthodes d'analyse et de caractérisation de la surface. Les formules permettant de calculer les dérivées partielles d'une surface *NURBS* sont données en annexe, à la section II.2.

### 1.2.2.3 Évaluation numérique des surfaces *NURBS*

Les méthodes d'évaluation sont très importantes dans ce projet en raison de l'utilisation intensive d'algorithmes d'optimisation évaluant un très grand nombre de fois la surface. De plus, comme nous le verrons au chapitre 3, la distance entre un point de l'espace et la surface s'exprime mathématiquement, elle aussi, sous la forme d'une fonction de type *NURBS*. Il est donc crucial d'évaluer rapidement une surface *NURBS*. L'algorithme récursif permettant d'évaluer un point sur une surface *NURBS* est fourni en annexe, à la section II.3.

### 1.2.2.4 Subdivision des surfaces *NURBS*

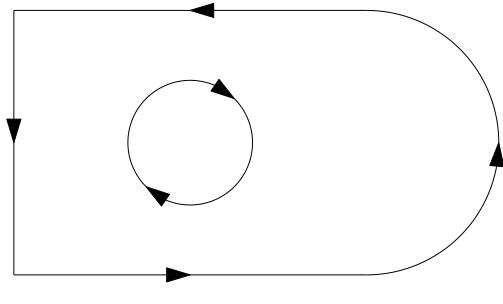
Une opération fondamentale sur les surfaces consiste à isoler une sous-surface d'une surface plus complexe. On cherche donc à subdiviser la surface en plusieurs sous-surfaces. La subdivision d'une surface *NURBS* utilise l'insertion de nœuds dans le vecteur nodal. En insérant  $p$  fois un nœud dans le vecteur nodal, on peut séparer la surface en deux sous-surfaces. Les détails des calculs permettant l'insertion des nœuds dans un vecteur nodal sont présentés en annexe, à la section II.4.

## 1.2.3 Surfaces de Bézier rationnelles

Les surfaces de Bézier rationnelles sont un cas particulier des surfaces *NURBS*. Afin de faciliter la manipulation et la caractérisation de surfaces gauches complexes modélisées à l'aide de *NURBS*, il est souvent nécessaire de subdiviser la surface en plusieurs carreaux de Bézier élémentaires. C'est cette approche que nous avons utilisée pour calculer le champ distance d'un solide représenté par une surface *NURBS*. Les détails des propriétés des surfaces de Bézier rationnelles ainsi que leur évaluation sont présentés à l'annexe III. Un point sur une surface de Bézier est calculé à l'aide de l'algorithme de De Casteljau. Cet algorithme est récursif et permet d'évaluer les polynômes exprimés dans une base de Bernstein. Les détails de l'algorithme sont fournis en annexe, à la section III.

#### 1.2.4 Orientation géométrique d'une surface paramétrique

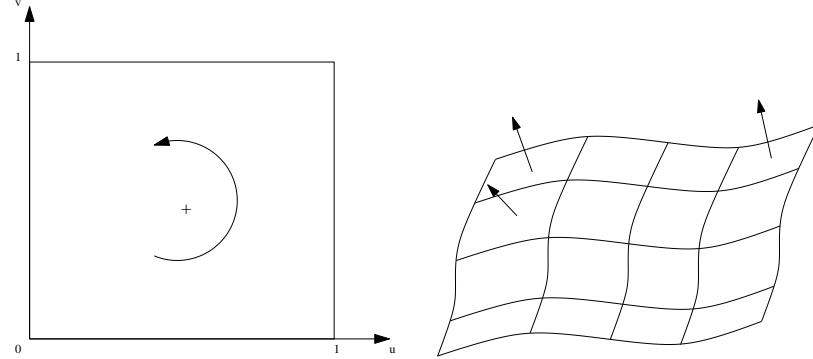
L'orientation géométrique dans un modèle topologique permet de distinguer l'intérieur de l'extérieur d'un solide. Le sens de déplacement indique l'orientation d'une courbe (voir Fig. 1.2). Par convention, l'intérieur est situé à gauche dans le sens de déplacement. Dans le cas des surfaces paramétriques, l'orientation est donnée



$$(x,y)=(x(s),y(s))$$

**Figure 1.2:** Orientation pour une représentation paramétrique.

grâce au sens de parcours des paramètres  $u$  et  $v$  (voir Fig. 1.3(a)). Par convention, l'orientation est calculée avec un produit vectoriel  $\vec{n} = \vec{u} \times \vec{v}$ ; l'orientation est donc liée au signe de la normal  $\vec{n}$  (voir Fig. 1.3(b)).



(a) Orientation du domaine paramétrique. (b) Orientation d'une surface paramétrique.

**Figure 1.3:** Orientation du domaine paramétrique et orientation résultante de la surface.

### 1.3 Modèle *B-Rep*

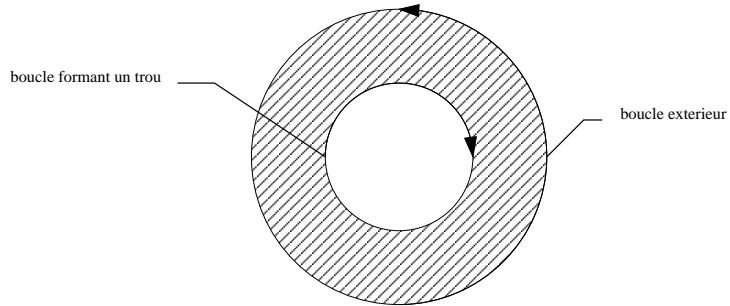
Lors de la modélisation d'un solide, il est important de choisir une méthode de représentation permettant de décrire, de traiter et de conserver efficacement sa géométrie. Le modèle de représentation *B-Rep* (abréviation de *Boundary Representation*) est très utilisé dans l'industrie pour la conception et la fabrication assistée par ordinateur. La représentation *B-Rep* consiste en une hiérarchie de morceaux de surface représentant la frontière du solide modélisé.

La cohérence de la représentation d'un solide nécessite de conserver deux types d'information :

- Les informations sur la topologie du solide qui donnent les relations de voisinage et de connexité entre trois types d'entités : les faces, les arêtes et les sommets.
- Les informations géométriques qui correspondent aux équations des courbes et des surfaces, ainsi que les coordonnées des points.

La représentation *B-Rep* impose, au niveau topologique, que chaque face représente une partie logique de la frontière de l'objet. Chacune de ces parties est représentée par un ensemble de boucles. Chacune des boucles est formée d'une liste d'arêtes, chaque arête étant définie par deux sommets. L'orientation des boucles est fondamentale pour orienter les faces afin de distinguer l'intérieur de l'extérieur du solide ainsi que les faces trouées. Les trous sont définis grâce au sens de parcours des arêtes (voir Fig. 1.4). Par convention, les boucles formant le contour extérieur du solide sont orientées selon le sens trigonométrique. Les boucles formant les trous sont orientées dans le sens anti-trigonométrique. Ainsi, lorsque l'on se déplace selon l'orientation d'une boucle, l'intérieur du solide est toujours situé *à gauche*.

L'ensemble des faces décrites dans la partie géométrique d'une représentation *B-Rep* doit représenter la totalité de la surface du solide modélisé. Afin que l'orientation topologique soit cohérente avec l'orientation géométrique, les équations des surfaces doivent également respecter les règles d'orientation et permettre de distinguer sans ambiguïté l'intérieur et l'extérieur du solide. Par convention, la normale en chaque point d'une surface doit être orientée vers l'extérieur du solide.



**Figure 1.4:** Orientation des boucles pour former un trou topologique.

## 1.4 Modèle de représentation implicite

Dans cette partie, nous décrirons les surfaces implicites et, plus particulièrement, une sous-famille définie à l'aide d'une fonction distance. Les surfaces implicites sont particulièrement adaptées à la représentation d'objets lisses ayant une topologie complexe. Il est simple de détecter, sur un modèle implicite, si un point donné de l'espace est à l'intérieur ou à l'extérieur de l'objet. Ce type de calcul est réalisable en temps réel, ce qui rend la modélisation par surface implicite particulièrement intéressante lors de la détection de collisions.

Nous dresserons donc un état de l'art des modèles de représentation implicite. Nous allons, dans un premier temps, donner une définition des surfaces implicites et une brève description des techniques de visualisation. Puis, dans un deuxième temps, nous expliquerons les différents algorithmes applicables à cette forme de modélisation.

### 1.4.1 Définition d'une surface implicite

Une surface implicite est définie à l'aide de deux entités :

1. une fonction  $f$ , appelée *fonction potentiel*, qui associe un réel à chaque point de l'espace ( $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ );
2. une valeur scalaire fixée, nommée *iso-valeur* et notée  $iso$ .

On obtient une surface en considérant l'ensemble des points  $S$  de l'espace  $\mathbb{R}^3$  pour lesquels la fonction potentielle  $f$  est égale à la valeur scalaire fixée  $iso$  :

$$S = \{P \in \mathbb{R}^3 / f(P) = iso\} .$$

Considérons trois exemples de surfaces définies de façon implicite. Un plan est défini de façon implicite comme :

$$\begin{aligned} f(x, y, z) &= ax + by + cz + d, \\ \text{Plan} &= \{P \in \mathbb{R}^3 / f(P) = 0\} . \end{aligned}$$

La surface d'une sphère centrée en  $(x_c, y_c, z_c)$  et de rayon  $R$  est définie sous la forme implicite :

$$\begin{aligned} f(x, y, z) &= (x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2, \\ \text{Sphère} &= \{P \in \mathbb{R}^3 / f(P) = R^2\} . \end{aligned}$$

D'une manière plus générale, une quadrique peut être définie de façon implicite comme :

$$\begin{aligned} f(x, y, z) &= [x, y, z, 1] \cdot Q \cdot [x, y, z, 1]^T, \\ \text{Quadrique} &= \{P \in \mathbb{R}^3 / f(P) = 0\} . \end{aligned}$$

où  $Q$  est la matrice caractéristique, de dimension  $4 \times 4$ , de la quadrique.

Les surfaces implicites divisent l'espace en deux régions, la région où  $f \geq iso$  et la région où  $f < iso$ . On définit l'intérieur de l'objet comme l'ensemble des points de l'espace dont la valeur de la fonction potentielle est inférieure à la valeur scalaire  $iso$ .

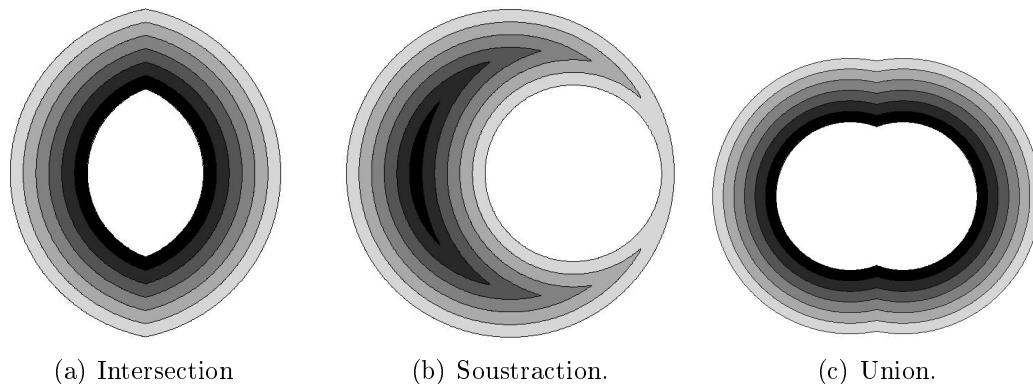
$$\begin{aligned} \text{intérieur} &= \{P \in \mathbb{R}^3 / f(P) < iso\} , \\ \text{extérieur} &= \{P \in \mathbb{R}^3 / f(P) \geq iso\} . \end{aligned}$$

Dans le cas du champ distance d'un solide, la géométrie originale du solide est représentée par la surface implicite dont l'iso-valeur est nulle. Un aspect important du champ distance d'un solide est sa capacité remarquable à représenter une surface située à une distance donnée de la surface originale. On appelle cette surface une *surface parallèle*. Une surface parallèle est constituée de l'ensemble des points de l'espace qui sont situés à une distance fixe du solide. Notons que la topologie d'une surface parallèle peut être différente de celle de la surface génératrice. L'ensemble de toutes les surfaces parallèles possibles d'un solide est contenu dans son champ distance.

#### 1.4.2 Opérations de construction de solides

Il est possible de construire de nouveaux objets à partir du champ distance de primitives de base (Payne et Toga, 1992). Soit deux champs distance orientés  $\phi_A$  et  $\phi_B$  correspondant à deux solides distincts représentés respectivement par leur frontière  $\Gamma_A$  et  $\Gamma_B$ . Les opérations booléennes sont définies comme :

$$\begin{aligned}\Gamma_A \cup \Gamma_B &= \min(\phi_A, \phi_B), \\ \Gamma_A \cap \Gamma_B &= \max(\phi_A, \phi_B), \\ \Gamma_A - \Gamma_B &= \max(\phi_A, -\phi_B).\end{aligned}$$



**Figure 1.5:** Opérations de construction de solides appliqués aux champs distance de deux cercles.

La figure 1.4.2 illustre différentes opérations booléennes à l'aide du champ distance de deux cercles.

### 1.4.3 Visualisation de surfaces implicites

En considérant la définition suivante d'une surface implicite  $S$  :

$$S = \{P \in \mathbb{R}^3 / f(P) = iso\} . \quad (1.2)$$

Il est légitime de s'interroger quant aux méthodes disponibles permettant de visualiser une telle construction mathématique. Indépendamment du mode de représentation, l'affichage d'une surface nécessite la localisation d'un ensemble de points. Déterminer des points appartenant à une surface implicite revient donc à localiser les points  $P$  de l'espace vérifiant l'équation 1.2.

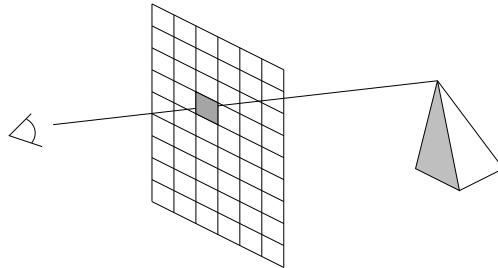
La principale difficulté réside dans la complexité de la fonction potentielle. Dans le cas des surfaces implicites, il est nécessaire de résoudre une équation, alors que les modèles paramétriques requièrent simplement l'évaluation d'une fonction analytique.

On distingue principalement trois méthodes pour visualiser les surfaces implicites : la méthode du lancer de rayon, le rendu discret et la facettisation de la surface implicite.

#### 1.4.3.1 Le lancer de rayon

La méthode de *lancer de rayon*, (aussi appelée *ray tracing*) permet de visualiser des surfaces implicites avec un rendu de très bonne qualité. Le lancer de rayon consiste à déterminer l'intersection entre un rayon partant de l'œil de l'observateur et la surface. Si un rayon rencontre la surface de l'objet, on allume le pixel situé sur la trajectoire entre l'observateur et la surface.

La méthode du lancer de rayon, bien que performante en termes de qualité de rendu graphique, reste très coûteuse en temps de calcul, car on doit résoudre une



**Figure 1.6:** Lancer de rayon. Le pixel situé sur un rayon partant de l'oeil de l'observateur et rejoignant la surface est allumé.

équation pour chaque pixel de l'image.

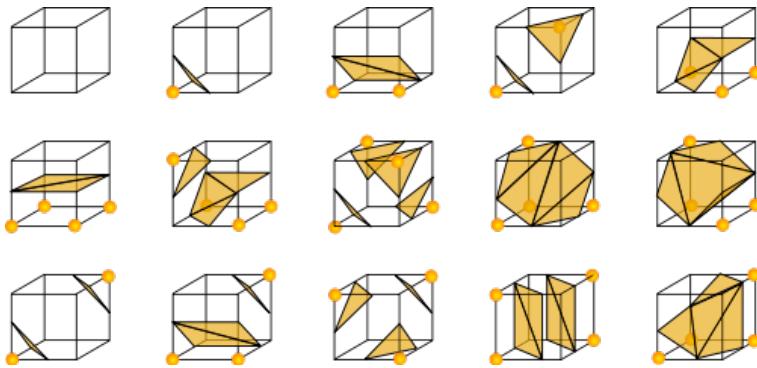
#### 1.4.3.2 Le rendu discret

Le méthode du *rendu discret* consiste à découper l'espace en petits volumes (cubiques) puis à effectuer un lancer de rayon sur la grille ou à projeter directement les cubes à l'écran. Cette méthode de visualisation est particulièrement adaptée dans les applications où la visualisation en temps réel est importante, par exemple pour l'affichage de données médicales où la projection orthogonale est préférée et où les champs scalaires sont généralement statiques. On renvoie le lecteur aux articles de Pfister (1999) et Mora *et al.* (2000).

#### 1.4.3.3 La facettisation

Les méthodes de *facettisation* consistent à construire une surface polygonale qui approxime la surface implicite. La méthode la plus connue est la méthode du *Marching Cubes* (Lorensen et Cline, 1987).

L'algorithme du *Marching Cubes* consiste d'abord à découper le domaine de définition de la surface implicite en une grille cartésienne tridimensionnelle. Puis, pour chaque cellule, on détermine la localisation des huit sommets en évaluant la fonction potentielle. Lorsque certains sommets de la cellule sont à l'intérieur de l'objet et d'autres à l'extérieur, alors la surface implicite coupe la cellule. Dans ce cas, , il faut générer les triangles adéquats pour approximer la surface implicite. La figure 1.7 présente les quatorze cas de base de l'algorithme du *Marching Cubes* en fonction



**Figure 1.7:** Les quatorze cas de base du l'algorithme du *Marching Cubes* (Lorensen et Cline, 1987).

du signe de la fonction potentielle aux différents sommets d'un cube élémentaire.

#### 1.4.4 Représentation paramétrique versus implicite

La représentation paramétrique et la représentation implicite sont complémentaires. Selon l'application, l'une des deux représentations peut être plus avantageuse. Le tableau 1.1 résume brièvement les avantages et les inconvénients de ces deux formes de représentation.

#### 1.4.5 Transformation vers une représentations implicite

La transformation d'une géométrie paramétrique en une géométrie implicite n'est pas une opération triviale. La conversion exacte est, pour un cas général, sans espoir (Velho et Gomes, 1996). Cependant si l'on abandonne l'idée d'une transformation exacte permettant de déterminer une forme implicite d'une représentation paramétrique, il est possible de développer une transformation approximative.

Le calcul de la carte de distance permet justement de passer d'une représentation paramétrique d'un solide quelconque à une représentation implicite. Le champ distance est une fonction potentielle  $\phi$  telle que  $|\phi(p)|$  est la plus courte distance euclidienne entre le point  $p$  et la surface d'un solide. Bien que le calcul du champ distance soit approximatif, il est possible d'obtenir un champ distance aussi précis

**Tableau 1.1:** Comparaison des représentations implicites et paramétriques (Patrikalakis, 2002)

Inconvénients	
Implicite	Paramétrique
<ul style="list-style-type: none"> <li>– Difficulté à déterminer la forme paramétrique d'un objet quelconque ;</li> <li>– Dépendance aux axes ;</li> <li>– Visualisation complexe de la surface ;</li> <li>– Difficulté à représenter précisément des arêtes vives.</li> </ul>	<ul style="list-style-type: none"> <li>– Intersections difficiles à calculer ;</li> <li>– Difficulté à localiser des points.</li> </ul>
Avantages	
Implicite	Paramétrique
<ul style="list-style-type: none"> <li>– Localisation facile des points ;</li> <li>– Intersections et offsets facilement représentables.</li> </ul>	<ul style="list-style-type: none"> <li>– Indépendance par rapport aux axes (transformation facile) ;</li> <li>– Possibilité d'interpoler des points pour former une surface ;</li> <li>– Visualisation facile.</li> </ul>

qu'on le souhaite en raffinant la grille cartésienne.

Les méthodes de calcul du champ distance actuellement disponibles ne s'appliquent qu'aux représentations polygonales. Comme nous l'expliquerons dans le chapitre suivant, le calcul des distances repose sur une propagation de conditions limites, lesquelles sont fortement liées à la représentation géométrique. Il serait donc souhaitable de calculer le champ distance d'une représentation paramétrique, qui est lisse, à l'aide des meilleures conditions limites possibles, afin d'obtenir une représentation implicite la plus précise possible. La construction de conditions limites à partir d'une discréttisation polygonale d'une géométrie paramétrique n'est pas souhaitable, car les conditions limites sont alors entachées d'une erreur de représentation due à la discréttisation.

## CHAPITRE 2

### CHAMP DISTANCE D'UN SOLIDE

Ce chapitre a pour principaux objectifs de formaliser le concept de champ distance d'un solide et de présenter une revue bibliographique des méthodes numériques de propagation de conditions limites pour le calcul d'une carte de distance.

Dans un premier temps, nous donnerons une définition du champ distance continu ainsi que du champ distance discret. Dans un deuxième temps, nous exposerons les différentes méthodes existantes permettant de calculer le champ distance discret d'une géométrie quelconque. En particulier, nous présenterons la formulation Eikonal de la fonction distance ainsi que les schémas numériques permettant de la résoudre. Finalement, nous décrirons les propriétés mathématiques d'un champ distance.

#### 2.1 Définition du champ distance d'un solide

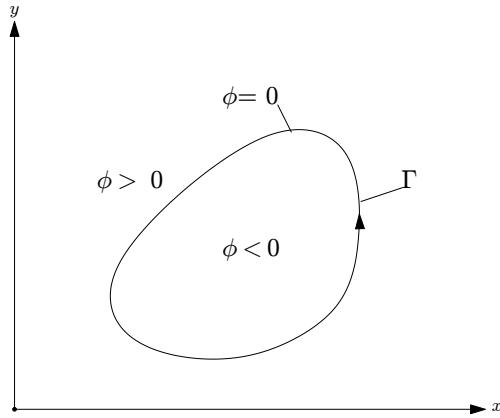
Un *champ distance*, appelé aussi *carte de distance*, est une représentation implicite d'un solide. La surface d'un solide est constituée de l'ensemble des points du champ distance dont la valeur est nulle. Le champ distance permet aussi de modéliser le volume entourant le solide.

Ainsi, le champ distance d'une surface fermée  $\Gamma$  est défini en tout point comme :  $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$  et pour  $p \in \mathbb{R}^3$ ,

$$\begin{aligned} \phi(p) &= \text{sgn}(p) \cdot \min\{|p - q| : q \in \Gamma\} \\ \text{sgn}(p) &= \begin{cases} +1 & \text{si } p \text{ extérieur,} \\ -1 & \text{si } p \text{ intérieur.} \end{cases} \end{aligned} \tag{2.1}$$

où  $|p - q|$  est la distance euclidienne entre le point  $p$  de l'espace et un point  $q$  appartenant à  $\Gamma$ . Une valeur absolue élevée signifie que l'on est situé loin de la

surface du solide. À contrario, une valeur faible indique que le point est proche de la surface. La figure 2.1 représente l'orientation du champ distance d'un parcours  $\Gamma$  simple est fermé.



**Figure 2.1:** Orientation du champ distance d'un parcours simple fermé.

De la même façon que toute hypersurface compacte de  $R^n$  est orientable, la carte de distance d'une surface fermée est orientée. Les valeurs du champ distance peuvent être positives ou négatives. Une valeur négative correspond à un point situé à l'intérieur du solide, tandis qu'une valeur positive correspond à un point situé à l'extérieur. Cette propriété est une généralisation du théorème des courbes de Jordan en dimension supérieure à deux.

Dans le cas d'une surface non orientée le champ distance est défini positif ou nul dans tout le domaine de définition.

Le champ distance peut être représenté de façon continue ou discontinue.

### 2.1.1 Champ distance continu

Un champ distance continu est une fonction analytique définie sur  $\mathbb{R}^n$ , donnant la distance euclidienne minimale permettant de rejoindre la surface  $\Gamma$  d'un solide.

Par exemple, le champ distance d'une sphère de rayon  $r$  et centrée en  $\vec{c}$  est donné

par la fonction analytique suivante :

$$\phi(\vec{x}) = \begin{cases} -|\vec{c} - \vec{x}|, & |\vec{c} - \vec{x}| < r, \\ 0, & |\vec{c} - \vec{x}| = r, \\ +|\vec{c} - \vec{x}|, & |\vec{c} - \vec{x}| > r. \end{cases}$$

Mis à part des formes géométriques simples, telles que certains solides de révolution comme une sphère, il est difficile, voir impossible, de déterminer une fonction analytique donnant la distance euclidienne minimale à la surface d'un solide quelconque. Il est donc nécessaire de considérer le problème du calcul du champ distance d'une géométrie quelconque dans un domaine discret, ainsi que la mise en oeuvre de méthodes de résolution numériques.

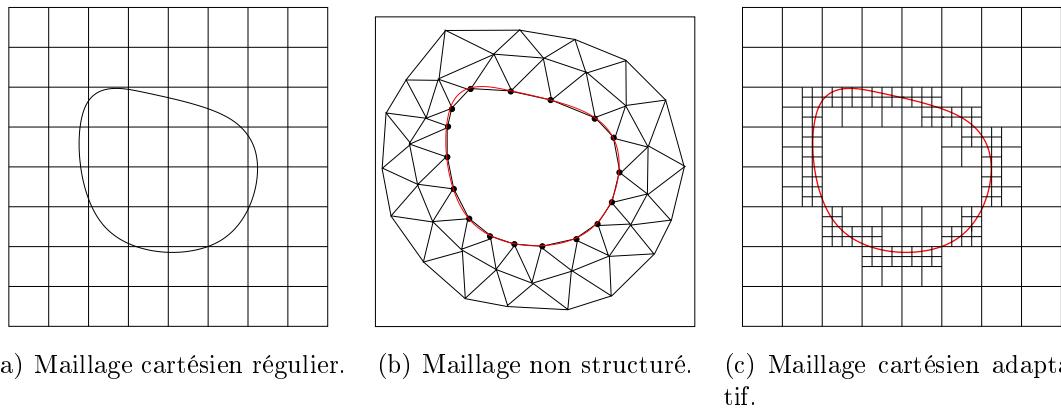
### 2.1.2 Champ distance discret

Il n'est pas toujours possible de représenter un champ distance sur  $\mathbb{R}^n$ ; on définit alors un sous-ensemble  $\Omega$  de  $\mathbb{R}^n$ . Le champ distance est défini dans le domaine de représentation borné  $\Omega$ .

**Définition 2.1.1 (champ distance discret)** *Un champ distance discret est représenté sur une grille de l'espace, définie sur  $\Omega$ , pour lesquel on connaît en chaque point la distance euclidienne minimale à la surface d'un solide.*

On définit une discrétisation de l'espace entourant le solide à l'aide d'un maillage structuré ou non structuré. Selon l'application et le type de discrétisation choisies, les méthodes utilisées pour calculer le champ distance seront différentes. Les maillages non structurés (illustrés à la figure 2.2(b)) sont utilisés pour les méthodes de calcul utilisant les éléments finis ou les méthodes de propagation frontales. Contrairement aux maillages cartésiens (illustrés à la figure 2.2(a) et 2.2(c)), les maillages non structurés exigent un savoir-faire pour générer un maillage de qualité représentant adéquatement la géométrie et donnant lieu à une bonne convergence

de la solution, permettant ainsi de résoudre correctement le transport de la frontière. De plus, un maillage non structuré du volume entourant un solide se raccroche à un maillage surfacique, lequel introduit une erreur de représentation de la géométrie. Nous reviendrons sur ce point à la section 2.6, où nous discuterons de la sensibilité aux conditions frontières.



**Figure 2.2:** Maillage pour le champ distance.

## 2.2 Méthodes de calcul du champ distance discret

Il existe de nombreuses méthodes permettant de calculer le champ distance d'un solide géométrique. La méthode la plus triviale consiste à utiliser une approche exhaustive. Pour chaque point du champ distance, on détermine le point de la surface le plus proche en calculant la projection orthogonale sur le solide. Cette méthode exhaustive est simple à programmer, mais est très coûteuse en temps de calcul. En effet, si on considère un champ distance discret de  $m$  points et une surface subdivisée en  $n$  éléments, le temps de calcul de la méthode exhaustive est en  $O(m \times n)$ . Bien que certaines méthodes de recherche géométrique et de localisation spatiale permettent de diminuer le nombre de comparaisons nécessaires, le temps de calcul de la méthode exhaustive demeure prohibitif pour un usage intensif (Satherley et Jones, 2001b).

Les méthodes de calcul du champ distance actuellement utilisées ne s'appliquent que pour des surfaces discrétisées. Aucune méthode n'a encore été proposée pour

calculer un champ distance directement à partir de surfaces paramétriques sans effectuer une discréétisation préalable. Les surfaces discréétisées sont représentées sous la forme d'un nuage de points ou d'un maillage surfacique de polygone tel que représenté à la figure 2.3. Le calcul du champ distance d'un solide devient ainsi fortement lié à la qualité du maillage surfacique.



**Figure 2.3:** Maillage surfacique d'une boîte de vitesse.

Le calcul de la carte de distance d'une géométrie repose sur le principe qu'il est possible de déterminer des conditions frontières proches de la surface permettant de diffuser l'information sur la distance euclidienne minimale à travers l'espace entourant le solide.

Le calcul de la transformée distance d'une géométrie nécessite deux étapes. La première consiste, à l'aide d'une méthode exhaustive, à déterminer des conditions frontières proches de la surface où l'on connaît exactement la distance minimale à la surface. La deuxième étape consiste à propager la distance à travers tout le champ distance. Ces deux étapes permettent de classifier les algorithmes permettant de calculer un champ distance selon deux critères. La méthode utilisée pour calculer une estimation locale de la distance minimale à partir des distances connues dans le voisinage immédiat forme le premier critère. La méthode utilisée pour propager la distance dans l'ensemble du domaine forme le second.

Le premier critère nous permet de répertorier trois types d'estimation locale de la distance minimale :

**Calcul par chanfrein.** Ces méthodes permettent de calculer la distance en un point à partir des points dans son voisinage, en additionnant une valeur à l'aide d'un patron. Le patron sous forme matriciel estime localement la distance (Butt et Maragos, 1998).

**Calcul vectoriel.** Ces méthodes conservent sous forme vectorielle le point de la surface qui est le plus proche. On détermine la valeur d'un point en appliquant un patron vectoriel à partir des points voisins (Satherley et Jones, 2001a).

**Équation Eikonal.** La valeur de la distance minimale en un point est approximée à l'aide d'un schéma de premier ordre à partir des points voisins (Tsitsiklis, 1995).

Les deux schémas de propagation relevant du second critère sont les suivants :

**Schéma de balayage.** Le schéma démarre dans un coin du domaine de calcul et propage l'information selon une ligne ou une colonne. Ce type de schéma nécessite plusieurs passages.

**Schéma d'avance de front.** Le schéma de propagation démarre à même la surface et avance par couches successives vers les frontières externes du domaine de calcul.

Les estimations locales, qui utilisent un patron scalaire ou vectoriel de propagation, sont imprécises. L'équation Eikonal permet une meilleure estimation locale de la distance à partir des conditions limites, et il est possible de développer des méthodes d'estimations d'ordre élevés. Pour ces raisons, dans le cadre de ce projet, nous nous intéressons à la formulation Eikonal du champ distance.

### 2.3 Équation Eikonal

L'équation Eikonal possède de nombreuses applications dans le domaine de la physique où elle permet notamment de faire le lien entre les théories de l'optique ondulatoire et de l'optique géométrique.

L'équation Eikonal est une équation non linéaire aux dérivées partielles définie par

la relation suivante :

$$|\nabla \phi(\vec{x})| = f(\vec{x}), \quad \vec{x} \in \mathbb{R}^n, \quad (2.2)$$

avec comme conditions limites la valeur exacte de  $\phi(\vec{x})$  pour  $\vec{x} \in \Gamma \subset \mathbb{R}^n$ . On cherche donc une fonction  $\phi(x)$  telle que la norme euclidienne de son gradient soit égale à  $f(\vec{x})$ .

La fonction  $f(\vec{x})$  est une fonction positive qui représente la vitesse de propagation de l'interface. Ainsi l'évaluation de la valeur de la fonction  $\phi(\vec{x})$  au point  $\vec{p}$  de l'espace donne le temps nécessaire au front de propagation pour atteindre le point  $\vec{p}$  à partir des conditions limites.

### 2.3.1 Formulation Eikonal de la fonction distance

On cherche à exprimer la distance euclidienne minimale entre un point et une surface sous la forme de l'équation Eikonal. Soit  $\vec{x}$  un point appartenant à une surface et  $\vec{P}$  un point de l'espace  $\mathbb{R}^n$ , on définit la fonction distance minimale  $\phi(\vec{x})$  comme :

$$\begin{aligned} \phi(\vec{x}) &= \left\| \vec{P} - \vec{x} \right\| \\ \phi(\vec{x}) &= \left( \vec{P} - \vec{x} \right) \cdot \left( \vec{P} - \vec{x} \right) \\ \phi(x_1, \dots, x_m) &= \sqrt{\sum_{n=1}^m (P_n - x_n)^2}. \end{aligned}$$

Les dérivées partielles de  $\phi(x_1, \dots, x_m)$  sont données par :

$$\frac{\partial \phi(\vec{x})}{\partial x_n} = \frac{P_n - x_n}{\sqrt{\sum_{n=1}^m (P_n - x_n)^2}}. \quad (2.3)$$

On peut écrire l'égalité suivante :

$$\begin{aligned}
 |\nabla\phi(\vec{x})|^2 &= \sum_{n=1}^m \left( \frac{\partial\phi(x_1, \dots, x_m)}{\partial x_n} \right)^2 \\
 &= \sum_{n=1}^m \left( \frac{P_n - x_n}{\sqrt{\sum_{n=1}^m (P_n - x_n)^2}} \right)^2 \\
 &= \frac{\sum_{n=1}^m (P_n - x_n)^2}{\sum_{n=1}^m (P_n - x_n)^2} \\
 |\nabla\phi(\vec{x})| &= 1.
 \end{aligned}$$

On peut obtenir le champ distance d'un solide en résolvant l'équation Eikonal telle que :

$$|\nabla\phi(\vec{x})| = 1. \quad (2.4)$$

avec comme conditions initiales la distance minimale à la surface pour les valeurs de  $\vec{x}$  qui forment la frontière du solide. Cette formulation indique que, dans tout l'espace, la vitesse de propagation est isotrope, ce qui traduit bien notre environnement.

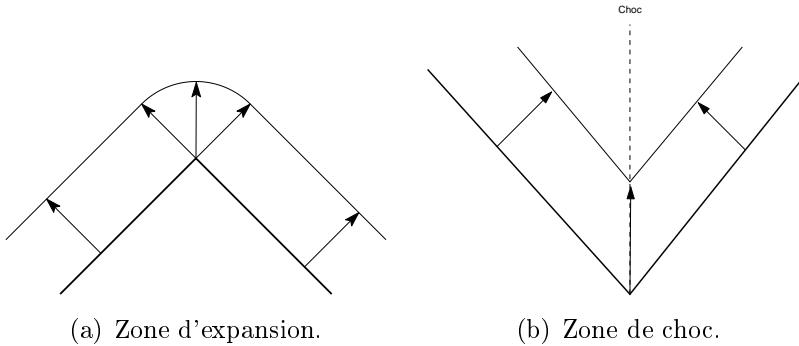
## 2.4 Intersection des caractéristiques

La résolution de l'équation Eikonal est un problème non linéaire aux valeurs frontières. C'est un problème aux dérivées partielles hyperbolique d'ordre un. L'information se propage le long des caractéristiques à partir de la frontière.

Il est possible que les caractéristiques ne couvrent pas une partie du domaine de l'équation aux dérivées partielles, formant ainsi une zone de raréfaction (appelée aussi zone d'expansion). Dans cette zone, la solution n'est connue que sous sa forme faible.

Lors de la propagation d'un coin, la normale n'est pas clairement définie, la direction

de propagation n'est donc pas unique (voir Fig. 2.4(a)). Il est donc nécessaire de recourir à une solution faible dans les zones de raréfaction.



**Figure 2.4:** Zone d'expansion engendrée par un sommet ((a)). L'information propagée depuis le sommet est insuffisante et amène une incertitude sur la direction de la normale. Dans le cas d'une courbure positive, les caractéristiques se rencontrent et on observe la formation d'un choc ((b)).

En raison de la non-linéarité des caractéristiques, on observe la formation de chocs à l'intersection des caractéristiques. La solution est continue dans ces singularités mais peut ne pas être différentiable. Krantz et Parks (1981) ont démontré que, pour une surface de continuité  $C^k$ , ( $k \geq 1$ ), la fonction distance continue et orientée est aussi de continuité  $C^k$ . La figure 2.4(b) illustre un cas où l'on observe la formation d'un choc.

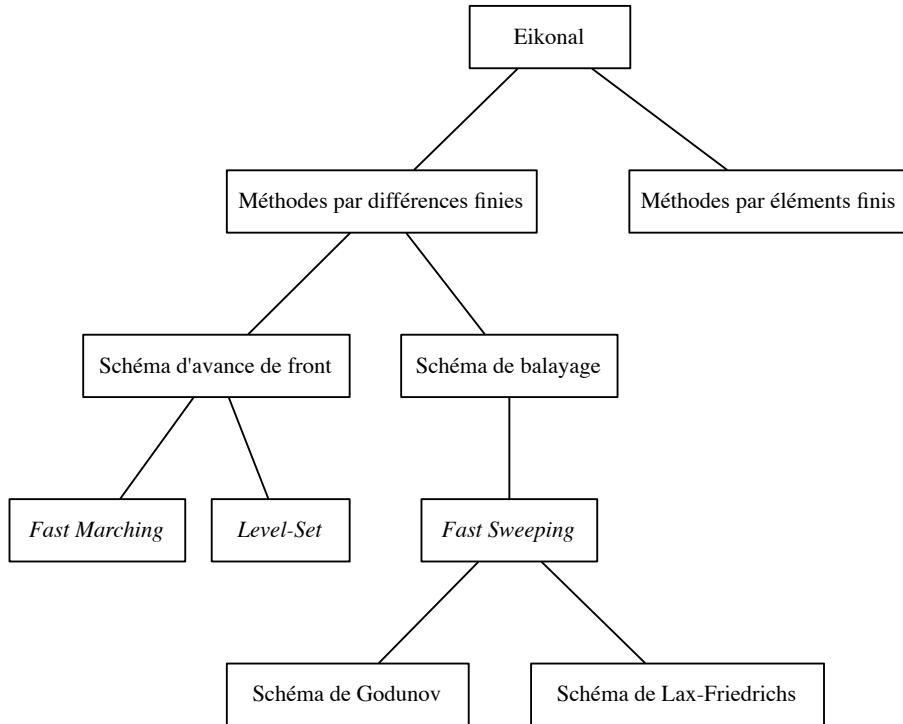
Le type de singularité formé dépend de la courbure locale de la surface. Une courbure positive provoque une intersection des caractéristiques, alors qu'une courbure négative génère une zone d'expansion.

## 2.5 Résolution de l'équation Eikonal

Nous nous intéressons aux méthodes de calcul du champ distance utilisant l'équation Eikonal. La résolution de cette équation est un problème non linéaire aux valeurs frontières. C'est un problème aux dérivées partielles hyperbolique d'ordre un. L'information se propage le long des caractéristiques à partir de la frontière. Le caractère non linéaire du problème conduit à la formation de chocs selon la loi

de conservation hyperbolique. La solution est toujours continue, mais peut ne pas être différentiable au niveau des chocs.

Zhao (2005) a démontré l'existence et l'unicité d'une solution à l'équation Eikonal.



**Figure 2.5:** Classification des méthodes de résolution de l'équation Eikonal.

L'équation Eikonal peut être résolue à l'aide de plusieurs méthodes. La figure 2.5 propose une classification des principales méthodes de résolution de l'équation Eikonal selon le schéma de propagation utilisé. Les principales méthodes seront décrites au cours de ce chapitre.

### 2.5.1 Méthodes de résolution par avance de front

Un premier ensemble de méthodes de résolution propage l'information selon un front qui se déplace dans la direction de propagation à chaque itération. Dans cette catégorie, on retrouve les méthodes de *Level-Set* et du *Fast Marching*.

### 2.5.1.1 Ensemble de niveaux ou *Level Set*

**Définition 2.5.1 (*Level Set*)** *Le level set d'une fonction dérivable  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  correspondant à un réel  $c$  est l'ensemble de points tels que :*

$$(x_1, \dots, x_n) \in \mathbb{R}^n : f(x_1, \dots, x_n) = c.$$

*Si  $n = 2$ , le level set est une courbe dans le plan appelée courbe de niveau. Si  $n = 3$  le levels set est une surface appelée surface de niveau.*

La méthode du *Level Set*, décrite dans Osher et Sethian (1988), est une technique simple et flexible permettant de calculer le déplacement d'une interface  $\Gamma$  en deux et trois dimensions. La frontière  $\Gamma$  délimite une région  $\Omega$  qui peut être non-simplement connexe. La frontière est propagée sous l'influence d'un champ de vitesse  $\vec{v}$ . La vitesse dépend de plusieurs facteurs tels que la position, le temps, la géométrie de l'interface et les conditions physiques externes. L'idée originale dans la méthode du *Level Set* est de définir une fonction  $\varphi(x, t)$  qui est régulière (*smooth*) et au minimum lipschitzienne<sup>1</sup> et qui représente implicitement l'interface en  $\varphi(x, t) = 0$ . La fonction *Level Set*  $\varphi$  a donc les propriétés suivantes :

$$\begin{cases} \varphi(x, t) > 0, & x \in \Omega, \\ \varphi(x, t) < 0, & x \ni \Omega, \\ \varphi(x, t) = 0, & x \in \partial\Omega = \Gamma(t). \end{cases}$$

Le déplacement de l'interface est calculé en propageant  $\varphi$  à travers le champ de vitesse  $\vec{v}$ . Cette propagation est calculée à l'aide de l'équation suivante :

$$\frac{\partial \varphi}{\partial t} + \vec{v} \cdot |\nabla \varphi| = 0.$$

La méthode du *Level Set* s'applique aux problèmes où le champ de vitesse peut être positif ou négatif, afin que le front de propagation puisse changer de direction.

---

<sup>1</sup>Une fonction lipschitzienne est une fonction possédant une régularité qui est plus forte que la continuité. Cette fonction est donc limitée dans sa manière d'évoluer.

La frontière peut alors se déplacer vers l'avant dans une zone et reculer dans une autre. Cette méthode est une généralisation de la méthode du Fast Marching où le champ de vitesse est constant, ce qui est le cas lors de la résolution de l'équation Eikonal.

### 2.5.1.2 *Fast Marching*

Le *Fast Marching* est une méthode numérique permettant de calculer le temps d'arrivée d'un front se déplaçant selon la normale à une courbe. On peut résoudre ce problème en déterminant la solution de l'équation Eikonal (Sethian, 1999). Le *Fast Marching* résoud donc le problème suivant :

$$|\nabla T| = \frac{1}{F}, \quad (2.5)$$

où  $F \geq 0$  est la vitesse de propagation du front et  $T$  son temps d'arrivée. Le *Fast Marching* utilise un schéma de discrétisation *upwind* permettant de résoudre cette équation :

$$\left( \begin{array}{l} \max(D_{ijk}^{-x}T, -D_{ijk}^{+x}T, 0)^2 \\ + \max(D_{ijk}^{-y}T, -D_{ijk}^{+y}T, 0)^2 \\ + \max(D_{ijk}^{-z}T, -D_{ijk}^{+z}T, 0)^2 \end{array} \right)^{1/2} = \frac{1}{F_{ijk}}. \quad (2.6)$$

Ce schéma de discrétisation permet de propager l'information dans une seule direction, des valeurs les plus petites vers les valeurs les plus élevées. Les valeurs sont propagées de proche en proche en ne considérant que les points qui sont contenus dans une bande proche des valeurs frontières. Les nouvelles valeurs obtenues sont fixées et ne changent plus lors des itérations suivantes. Ces valeurs deviennent alors les nouvelles valeurs frontières. La principale difficulté dans la méthode du *Fast Marching* réside dans la conception d'une méthode efficace permettant de déterminer la nouvelle bande de points à mettre à jour.

Voici les étapes d'une itération de la méthode du *Fast Marching* :

1. Marquer les points de la grille qui appartiennent à la frontière comme étant des valeurs connues ;
2. Marquer les points de la grille qui sont à côté des points connus comme étant

membres de la bande à mettre à jour ;

3. Marquer les points restant comme étant très éloignés ;
4. Calculer à l'aide de l'équation 2.6 la nouvelle valeur des points de la bande.

Les points de la bande devant être mis à jour sont déterminés à l'aide d'un tri de type *heap sort*. La méthode de tri, dont la complexité est en  $O(m \log m)$  est l'opération la plus lente. C'est l'opération limitante dans la méthode du *Fast Marching*.

### 2.5.2 Résolution par éléments finis

L'équation Eikonal peut être résolue à l'aide d'une méthode par éléments finis. Elias *et al.* (2007) proposent une méthode de calcul du champ distance sur un maillage non-structuré à l'aide d'un résolueur par éléments finis. L'équation Eikonal est imposée au niveau des éléments. Une nouvelle stratégie d'avance de front, basée sur le *Fast Marching*, est proposée. Cet algorithme utilise une liste d'éléments disponibles pour calculer la distance. Cette liste est mise à jour à chaque itération, en insérant, en activant et en désactivant certains noeuds pour l'itération suivante. Cette méthode est précise proche de la surface, et permet de limiter le calcul du champ distance à une bande entourant le solide.

Barth et Sethian (1998) considèrent une approche *stabilized space-time Galerkin least-squares*. Cette méthode par éléments finis permet de résoudre et de capturer les chocs des équations du type Hamilton-Jacobi et Eikonal.

Sussman et Fatemi (1999) résolvent les équations de *level-set* en utilisant une formulation *Streamline-Upwind/Petrov-Galerkin (SUPG)*. D'autres techniques permettant de résoudre les équations du *level-set* à l'aide des éléments finis ont aussi été développées, comme la méthode proposée par Mourad *et al.* (2005) qui utilise une formulation à gradient, ou encore la méthode de Marchandise *et al.* (2006) qui utilise une formulation de Galerkin.

### 2.5.3 Méthodes de résolution par balayage

Une deuxième catégorie de méthodes de résolution de l'équation Eikonal regroupe l'ensemble des méthodes utilisant un balayage de Gauss-Seidel.

#### 2.5.3.1 *Fast Sweeping*

Sous la dénomination *Fast Sweeping*, on fait référence à un ensemble de méthodes numériques permettant de résoudre les problèmes prenant la forme d'une équation de Hamilton-Jacobi. Le schéma itératif a la propriété fondamentale de converger en un nombre fini et connu de balayages.

Avant de décrire le fonctionnement du *Fast Sweeping* nous allons donner la définition d'une propriété de causalité, d'une caractéristique, et d'un schéma *upwind*.

**Définition 2.5.2 (propriété de causalité)** *Une propriété de causalité signifie que la valeur d'un point ne dépend que des valeurs de ses voisins. Au niveau de la résolution d'une équation, une propriété de causalité implique que la solution d'un point est uniquement influencée par son voisinage immédiat. On peut donc déterminer une solution localement.*

**Définition 2.5.3 (Caractéristiques)** *La méthode des caractéristiques est une méthode permettant de résoudre les équations aux dérivées partielles. Elle est principalement utilisée dans les domaines tels que la mécanique des fluides ou les problèmes de transport. Une caractéristique est, en deux dimensions, une courbe permettant de transformer une équation aux dérivées partielles de premier ordre en un système d'équations différentielles ordinaires. La résolution de l'équation différentielle ordinaire le long d'une caractéristique permet de résoudre le problème original. L'intersection des caractéristiques forme des chocs. La direction d'une caractéristique indique le sens de l'écoulement de la solution. La connaissance de cette direction permet de choisir une méthode de différence finie adaptée au problème.*

**Définition 2.5.4 (schéma *upwind*)** *En dynamique des fluides, un schéma *upwind* est une discrétisation qui permet de résoudre numériquement un problème*

hyperbolique aux dérivées partielles (comme l'équation d'onde, l'équation d'Euler). Les schémas upwind utilisent un patron de différences finies permettant de suivre la direction de propagation de l'information.

Le principe de base de la méthode du *Fast Sweeping* est d'utiliser une différence finie *upwind* non linéaire et des itérations de Gauss-Seidel suivant un ordre alterné. Contrairement à la méthode du *Fast Marching*, le *Fast Sweeping* suit les causalités selon des caractéristiques ayant des directions parallèles. Toutes les caractéristiques sont divisées en groupes selon leur direction, et chaque itération de Gauss-Seidel couvre un groupe de caractéristiques particulier. Les itérations de Gauss-Seidel sont effectuées selon un ordre spécifique permettant la propagation de toutes les caractéristiques du problème.

Dans le cas général de l'équation Eikonal (2.2), les caractéristiques ont un nombre infini de directions qui ne peuvent pas être exactement représentées par une grille cartésienne. Cependant, dans le cas particulier de la fonction distance 2.4, toutes les caractéristiques peuvent être classifiées en un nombre déterminé de groupes. Par exemple, en deux dimensions, toutes les directions des caractéristiques peuvent être classifiées en quatre groupes : en montant vers la droite, en montant vers la gauche, en descendant vers la gauche et en descendant vers la droite (voir Fig. 2.6).

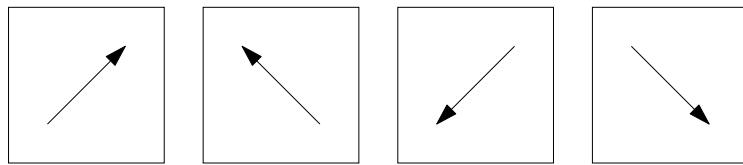
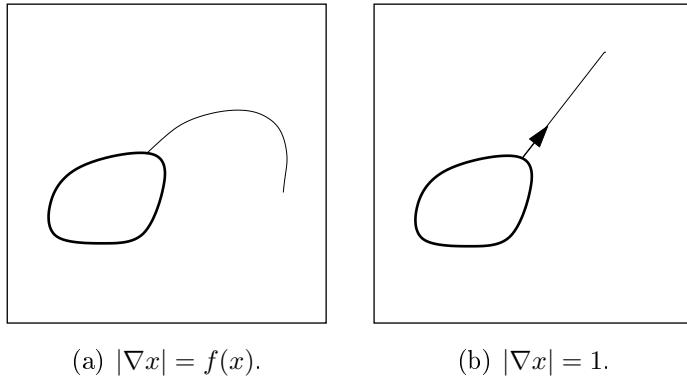


Figure 2.6: Directions du balayage en deux dimensions.

Pour l'équation Eikonal (2.2) les caractéristiques sont des courbes. Dans le cas particulier de la fonction distance (2.4), les caractéristiques forment des demi-droites orientées selon la normale à la surface.

On ne peut pas transporter une surface à l'aide des informations en aval. On doit donc utiliser les valeurs en amont pour les propager. Les schémas *upwind* permettent de suivre la caractéristique et de propager l'information vers l'avant.



**Figure 2.7:** Caractéristiques de l'équation Eikonal.

Plusieurs méthodes utilisent un schéma de propagation du type *Fast Sweeping* pour résoudre l'équation Eikonal. Il existe aussi plusieurs schémas de discrétisation de l'équation Eikonal. Parmi ceux-ci, on distingue le schéma de Lax-Friedrichs et celui de Godunov.

#### 2.5.4 Schéma de Lax-Friedrichs

Un schéma de Lax-Friedrichs est proposé dans Kao *et al.* (2004). Il permet de résoudre les problèmes prenant la forme d'une équation de Hamilton-Jacobi statique :

$$\begin{cases} H(x, \nabla \phi(x)) = R(x) & x \in \Omega, \\ \phi(x) = q(x) & x \in \Gamma. \end{cases}$$

Les fonctions  $H$ ,  $q$ , et  $R > 0$  sont lipschitziennes.  $\Gamma$  est un sous ensemble de  $\Omega$  pour lequel les valeurs exactes de la solution sont connues. Cette méthode peut résoudre aussi bien les Hamiltoniens convexes que non convexes.

Pour faciliter la compréhension, nous allons illustrer le schéma de Lax-Friedrichs dans le cas unidimensionnel. L'approximation de Lax-Friedrichs de l'Hamiltonien est :

$$\tilde{H}^{LF}(p^-, p^+) = H\left(\frac{p^- + p^+}{2}\right) - \sigma_x\left(\frac{p^- - p^+}{2}\right),$$

où  $p = \frac{\partial \phi}{\partial x}$  et  $p^\pm$  correspondent aux différences finies en amont et en aval de l'approximation de  $p$ . La constante  $\sigma_x$  est une viscosité artificielle vérifiant :

$$\sigma_x \geq \max_{p \in [A, B]} \left| \frac{\partial H}{\partial p} \right|.$$

Numériquement, en considérant une discrétisation uniforme  $\{x_i, i = 1, \dots, m\}$  de taille  $\delta_x$  du domaine  $\Omega$ , on peut poser par approximation, en chaque point  $x_i$ , que l'égalité suivante est vérifiée :

$$\tilde{H}^{LF} = R,$$

ce qui implique

$$H \left( x_i, \frac{\phi_{i+1} - \phi_{i-1}}{2\delta x} \right) - \sigma_x \left( \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{2\delta x} \right) = R(x_i).$$

À chaque itération, les nouvelles valeurs sont calculées à l'aide du schéma suivant :

$$\begin{aligned} \phi_{i,j,k}^{n+1} = c \left\{ R - H \left( \frac{\phi_{i+1,j,k} - \phi_{i-1,j,k}}{2\Delta x}, \frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2\Delta y}, \frac{\phi_{i,j,k+1} - \phi_{i,j,k-1}}{2\Delta z} \right) \right. \\ \left. + \sigma_x \frac{\phi_{i+1,j,k} + \phi_{i-1,j,k}}{2\Delta x} + \sigma_y \frac{\phi_{i,j+1,k} + \phi_{i,j-1,k}}{2\Delta y} + \sigma_z \frac{\phi_{i,j,k+1} + \phi_{i,j,k-1}}{2\Delta z} \right\}, \end{aligned} \quad (2.7)$$

avec

$$c = \frac{1}{\frac{\sigma_x}{\Delta_x} + \frac{\sigma_y}{\Delta_y} + \frac{\sigma_z}{\Delta_z}},$$

où  $\sigma_x$ ,  $\sigma_y$  et  $\sigma_z$  représentent des viscosités artificielles déterminées en fonction du modèle de propagation désirée. Le schéma de résolution utilise une discrétisation sur une grille cartésienne en 3D. Les valeurs  $\Delta_{x\dots z}$  représentent la taille des éléments dans chacune des directions spatiales.

Contrairement à un Hamiltonien de Godunov, un Hamiltonien de Lax-Friedrichs n'est pas un schéma *upwind*. Étant donné que les caractéristiques peuvent se propager vers l'extérieur du domaine de calcul, un Hamiltonien de Godunov serait capable de calculer correctement les points situés sur la frontière du domaine de calcul. Par contre, comme un Hamiltonien de Lax-Friedrichs prend en compte tous

les points de son voisinage, il faut impérativement spécifier les points situés en dehors du domaine de calcul, sinon on introduit des erreurs importantes dans la solution qui vont se propager à l'intérieur du domaine de calcul. L'imposition des conditions frontières aux bords du domaine est calculée à l'aide des équations suivantes :

$$\begin{cases} \phi_{0,j}^{new} &= \min(\max(2\phi_{1,j} - \phi_{2,j}, \phi_{2,j}), \phi_{0,j}^{old}), \\ \phi_{m_1+1,j}^{new} &= \min(\max(2\phi_{m_1,j} - \phi_{m_1-1,j}, \phi_{m_1-1,j})\phi_{m_1+1,j}^{old}), \\ \phi_{i,0}^{new} &= \min(\max(2\phi_{i,1} - \phi_{i,2}, \phi_{i,2}), \phi_{i,0}^{old}), \\ \phi_{i,m_2+1}^{new} &= \min(\max(2\phi_{i,m_2} - \phi_{i,m_2-1}, \phi_{i,m_2-1})\phi_{i,m_2+1}^{old}). \end{cases} \quad (2.8)$$

L'algorithme basé sur un schéma de Lax-Friedrichs se déroule en quatre étapes :

1. **Initialisation.** La valeur exacte de  $\phi_{i,j}^0$  est assignée aux points de la grille qui sont sur ou proche de la frontière  $\Gamma$ . Ces valeurs sont fixées pour toutes les itérations suivantes. On assigne à tous les autres points de la grille une valeur  $M$  positive et élevée. La valeur de  $M$  doit être plus grande que la plus grande des valeurs possibles de la solution.
2. **Balayages alternés.** À l'itération  $n+1$ , on calcule la nouvelle valeur de  $\phi_{i,j}^{n+1}$  en utilisant la formule 2.7. La valeur de  $\phi_{i,j}^n$  est remplacée par la nouvelle valeur  $\phi_{i,j}^{n+1}$  si et seulement si elle est inférieure à  $\phi_{i,j}^n$ . En deux dimensions, quatre balayages dans des directions alternées sont effectués pour propager l'information :
  - (a) du coin inférieur gauche vers le coin supérieur droit ;
  - (b) du coin inférieur droit vers le coin supérieur gauche ;
  - (c) du coin supérieur droit vers le coin inférieur gauche ;
  - (d) du coin supérieur gauche vers le coin inférieur droit.
3. **Imposition des conditions frontières.** Après chaque itération, on impose les conditions frontières à l'aide de la formule 2.8.
4. **Vérification de la convergence.** Soit un critère de convergence  $\epsilon > 0$ . On vérifie que :
$$\|\phi^{n+1} - \phi^n\|_{L_1} \leq \epsilon.$$

Dans le cas particulier de la résolution de l'équation distance, il est préférable d'utiliser un schéma *upwind* afin de propager la surface. Ainsi, nous devons seulement connaître les conditions frontières des points situés autour du solide.

Dans le cas particulier du calcul d'un champ distance, il est plus difficile d'initialiser les conditions limites lorsqu'elles sont éloignées du solide. Pour cette raison, nous avons choisi d'utiliser le schéma *upwind* de Godunov proposé par Zhao, afin de résoudre l'équation distance. Ainsi, nous n'avons pas besoin de calculer les conditions limites aux bords du domaine de calcul.

### 2.5.5 Schéma de Godunov

Une méthode de résolution de l'équation Eikonal généralisée, appelée le *Fast Sweeping*, est proposée par Zhao (2005).

#### 2.5.5.1 Description de la méthode du *Fast Sweeping*

La méthode du *Fast Sweeping* permet de calculer la solution de viscosité  $u(\mathbf{x}) \geq 0$  de l'équation Eikonal généralisée suivante :

$$\begin{aligned} |\nabla u(\vec{x})| &= f(\vec{x}) & \vec{x} \in R^n, \\ u(\vec{x}) &= 0 & \vec{x} \in \Gamma \in R^n, \end{aligned} \tag{2.9}$$

où  $f(\vec{x}) > 0$ . On note  $x_{i,j}$  un nœud de la grille faisant partie du domaine de calcul  $\Omega$  et  $u_{i,j}^h$  est la solution au nœud  $x_{i,j}$ . Le domaine de calcul  $\Omega$  est discrétisé à l'aide d'une grille cartésienne régulière de taille  $h$ .

La méthode du *Fast Sweeping* utilise un schéma de Godunov *upwind* afin de discrétiser l'équation aux dérivées partielles Eikonal. Pour une grille cartésienne de dimension  $I \times J$ , la discrétisation de l'équation 2.2 est :

$$[(u_{i,j}^h - u_{xmin}^h)^+]^2 + [(u_{i,j}^h - u_{ymin}^h)^+]^2 = f_{i,j}^2 h^2, \tag{2.10}$$

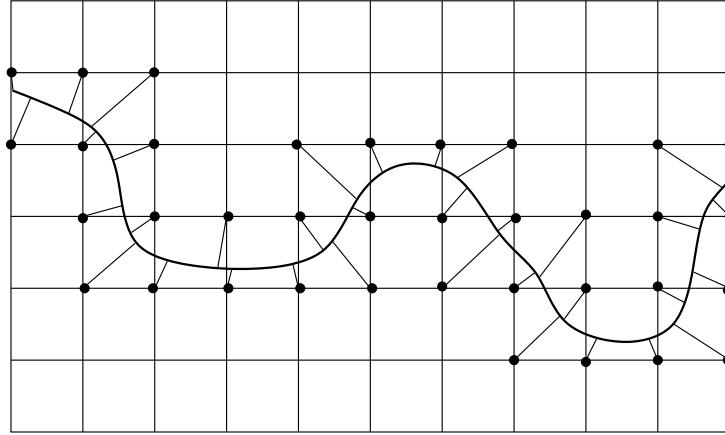
avec  $i = 2, \dots, I - 1$  et  $j = 2, \dots, J - 1$ . On définit les valeurs  $u_{xmin}^h$  et  $u_{ymin}^h$  tel que :

$$\begin{aligned} u_{xmin}^h &= \min(u_{i-1,j}^h, u_{i+1,j}^h), \\ u_{ymin}^h &= \min(u_{i,j-1}^h, u_{i,j+1}^h). \end{aligned}$$

La fonction échelon  $(x)^+$  est définie comme :

$$(x)^+ = \begin{cases} x & x > 0, \\ 0 & x \leq 0. \end{cases}$$

Les conditions frontières  $u(x) = 0$ , pour  $x \in \Gamma \subset \mathbb{R}^n$ , sont imposées aux nœuds de la grille qui sont les plus proches ou qui sont situés sur la frontière  $\Gamma$ . On assigne aux autres nœuds de la grille une valeur très grande par rapport au problème.



**Figure 2.8:** Représentation d'une géométrie  $\Gamma$  et des nœuds de la grille cartésienne formant les conditions frontières. Les distances minimales utilisées pour l'initialisation des conditions frontières sont illustrées par des segments.

Le schéma de différence finie *upwind* utilisé pour discréteriser l'équation Eikonal respecte la propriété de causalité suivante : la solution d'un nœud de la grille est uniquement déterminée par ses voisins immédiats ayant une valeur inférieure. Le schéma de discréttisation décentré (*one-sided*) aux frontières permet de propager l'information de l'intérieur vers l'extérieur.

Une fois les conditions frontières imposées sur la grille cartésienne, l'équation 2.10 est résolue à l'aide de plusieurs itérations de Gauss-Seidel en alternant la direction du balayage lexicographique.

La solution de l'équation Eikonal en deux dimensions :

$$[(x - a)^+]^2 + [(x - b)^+]^2 = f_{i,j}^2 h^2, \quad (2.11)$$

où  $a = u_{xmin}^h$ ,  $b = u_{ymin}^h$  est

$$\bar{x} = \begin{cases} \min(a, b) + f_{i,j} h & |a - b| \geq f_{i,j} h, \\ \frac{a + b + \sqrt{2f_{i,j}^2 h^2 - (a - b)^2}}{2} & |a - b| < f_{i,j} h. \end{cases}$$

En trois dimensions, l'équation Eikonal est discrétisée de la façon suivante :

$$[(x - a)^+]^2 + [(x - b)^+]^2 + [(x - c)^+]^2 = f_{i,j}^2 h^2. \quad (2.12)$$

Les dérivées partielles sont approximées à l'aide des différences finies suivantes :

$$\begin{aligned} \frac{\partial \phi}{\partial x} &= \frac{\phi_{i,j,k}^{t+1} - a}{h}, \\ \frac{\partial \phi}{\partial y} &= \frac{\phi_{i,j,k}^{t+1} - b}{h}, \\ \frac{\partial \phi}{\partial z} &= \frac{\phi_{i,j,k}^{t+1} - c}{h}. \end{aligned} \quad (2.13)$$

Pour une grille cartésienne de dimension  $I \times J \times K$ , les valeurs des coefficients  $a, b$  et  $c$  sont :

$$\begin{aligned} a &= \min(\phi_{i-1,j,k}^t, \phi_{i+1,j,k}^t) \quad i = 2, \dots, I-1, \\ b &= \min(\phi_{i,j-1,k}^t, \phi_{i,j+1,k}^t) \quad j = 2, \dots, J-1, \\ c &= \min(\phi_{i,j,k-1}^t, \phi_{i,j,k+1}^t) \quad k = 2, \dots, K-1. \end{aligned} \quad (2.14)$$

La nouvelle valeur calculée à chaque itération est obtenue en utilisant la formule :

$$\phi^{t+1} = \frac{(a + b + c) + \sqrt{3h^2 + 2(ab + bc + ac - a^2 - b^2 - c^2)}}{3}. \quad (2.15)$$

**2.5.5.1.1 Algorithme du *Fast Sweeping*** La méthode du *Fast Sweeping* se décompose en trois étapes. En premier lieu, on détermine les nœuds formant les conditions limites. En deuxième lieu, on les initialise. En troisième lieu, on propage la frontière à l'aide de plusieurs balayages :

1. **Détection des nœuds frontières.** On détermine les nœuds de la grille cartésienne qui sont les plus proches de la géométrie. Ces nœuds formeront les conditions frontières permettant de résoudre l'équation Eikonal.
2. **Initialisation des conditions frontières.** On initialise les nœuds frontières avec la valeur exacte de  $\phi$ , c'est-à-dire la distance euclidienne minimale à la surface.
3. **Balayage.** À l'itération  $n+1$ , on calcule la nouvelle valeur de  $\phi_{i,j}^{n+1}$  en utilisant la formule 2.15. La valeur de  $\phi_{i,j}^n$  est remplacée par la nouvelle valeur  $\phi_{i,j}^{n+1}$  si et seulement si elle est inférieure à  $\phi_{i,j}^n$ . En deux dimensions, quatre balayages dans des directions alternées sont effectués pour propager l'information :
  - (a) du coin inférieur gauche vers le coin supérieur droit ;
  - (b) du coin inférieur droit vers le coin supérieur gauche ;
  - (c) du coin supérieur droit vers le coin inférieur gauche ;
  - (d) du coin supérieur gauche vers le coin inférieur droit.

### 2.5.5.2 Convergence du *Fast Sweeping*

Il n'est pas possible de déterminer le nombre d'itérations nécessaire permettant d'obtenir la convergence d'un système hautement non linéaire. Cependant, Zhao a démontré que pour un problème de dimension  $n$ , il suffisait d'effectuer  $2^n$  itérations de la méthode du *Fast Sweeping* afin d'obtenir une solution de la fonction distance aussi précise que la solution obtenue après convergence. On entend par

"aussi précise" que l'erreur de la solution après  $2^n$  itérations est en  $O(h)$ , c'est-à-dire inférieure à la taille de la discréétisation choisie. Par conséquent, d'après les travaux de Zhao,  $2^n$  itérations du *Fast Sweeping* sont en pratique suffisantes pour calculer la solution de la fonction distance pour une géométrie quelconque. En trois dimensions, il est ainsi nécessaire d'effectuer huit balayages.

### 2.5.5.3 Propriétés du *Fast Sweeping*

Nous allons présenter les propriétés les plus importantes de la méthode du *Fast Sweeping*. Les preuves et les démonstrations sont dans Zhao (2005).

**Lemme 2.5.5** *Soit  $\bar{x}$  la solution de l'équation 2.9, nous avons :*

$$1 \geq \frac{\partial \bar{x}}{\partial a_1} \geq \frac{\partial \bar{x}}{\partial a_2} \geq \dots \geq \frac{\partial \bar{x}}{\partial a_n} \geq 0,$$

et

$$\frac{\partial \bar{x}}{\partial a_1} + \frac{\partial \bar{x}}{\partial a_2} + \dots + \frac{\partial \bar{x}}{\partial a_n} = 1.$$

**Lemme 2.5.6** *À chaque itération de Gauss-Seidel, la solution est strictement décroissante.*

**Corollaire 2.5.7** *Soit  $u^{(k)}$  et  $v^{(k)}$  deux solutions numériques à la  $k^e$  itération du Fast Sweeping. On note  $\|\cdot\|_\infty$  la norme maximale. Nous avons les deux propriétés suivantes :*

$$\|u^{(k)} - v^{(k)}\|_\infty \leq \|u^{(k-1)} - v^{(k-1)}\|_\infty,$$

et

$$0 \leq \max \left( u_{i,j}^{(k)} - u_{i,j}^{(k+1)} \right) \leq \max \left( u_{i,j}^{(k-1)} - u_{i,j}^{(k)} \right).$$

Avant que l'information provenant de la frontière  $\Gamma$  ait atteint tous les nœuds de la grille,  $\|u^{(k)} - u^{(k-1)}\|_\infty$  est de l'ordre  $O(1)$ . Quand  $\|u^{(k)} - u^{(k-1)}\|_\infty$  est de l'ordre de  $O(h)$ , l'information a été propagée à tous les nœuds de la grille et on

peut arrêter le processus itératif. Bien que la solution n'ait pas encore convergée, la solution numérique est aussi précise que si on avait atteint la convergence. La solution itérative va changer de moins en moins à chaque itération et va converger graduellement vers la solution du système discrétisé.

**Théorème 2.5.8** *La solution itérative obtenue en utilisant la méthode du Fast Sweeping converge de façon monotone vers la solution du système discrétisé (Zhao, 2005).*

#### 2.5.5.4 *Fast Sweeping* d'ordre supérieur

Zhang *et al.* (2006) une méthode numérique d'ordre supérieur est proposé. Cette méthode permet de calculer la solution de viscosité<sup>2</sup> de l'équation Eikonal sur une grille cartésienne régulière en utilisant une approximation *weighted essentially non-oscillatory* (WENO) des dérivées ainsi que des itérations de Gauss-Seidel selon un ordre alterné.

#### 2.5.6 Choix de la méthode retenue

Comme nous l'avons expliqué précédemment, plusieurs schémas de résolution sont disponibles pour calculer le champ distance d'un solide avec l'équation Eikonal. Cependant, toutes ces méthodes ne sont pas équivalentes en termes de précision et de performance. Les méthodes par éléments finis tout comme la méthode du *Fast-Marching* nécessitent la génération d'un maillage et donc une discrétisation de la géométrie. Comme nous l'expliquerons dans la prochaine section, l'erreur des conditions limites se propage intégralement dans tout le champ distance. Nous voulons ainsi éviter d'utiliser une méthode qui nécessite de discréter la représentation paramétrique, afin d'obtenir des conditions frontières les plus précises possible.

---

<sup>2</sup>La notion de solution de viscosité est une notion de solution faible pour les équations elliptiques ou paraboliques non linéaires dégénérées. Cette notion a démontré son efficacité dans des domaines d'applications tels que les mouvements géométriques et de propagation d'ondes.

En termes de performance, Gremaud et Kuster (2006) a montré que le *Fast-Sweeping* est plus rapide que le *Fast-Marching* lorsque le changement de direction du gradient n'est pas trop élevé et lorsque que le front de propagation est important. C'est le cas lors du calcul d'un champ distance : le front de propagation est important car il est formé par toute la surface du solide et le changement de direction du gradient ne survient qu'à l'intersection des caractéristiques. Nous avons donc choisi la méthode de propagation du *Fast-Sweeping* pour calculer le champ distance.

En ce qui a trait à l'estimation locale et à la discréétisation de l'équation Eikonal, nous avons le choix entre le schéma de Lax-Friedrichs et celui de Godunov. L'initialisation des conditions frontières pour le calcul du champ distance d'une géométrie paramétrique repose sur la projection d'un point sur une surface. Pour des raisons de précision et de performance, nous ne voulons pas projeter des points qui sont loin de la surface. Comme le schéma de Lax-Friedrichs nécessite d'initialiser les conditions frontières aux bords du domaine de calcul, nous avons choisi le schéma Godunov qui ne requiert qu'une initialisation de conditions limites proches de la surface.

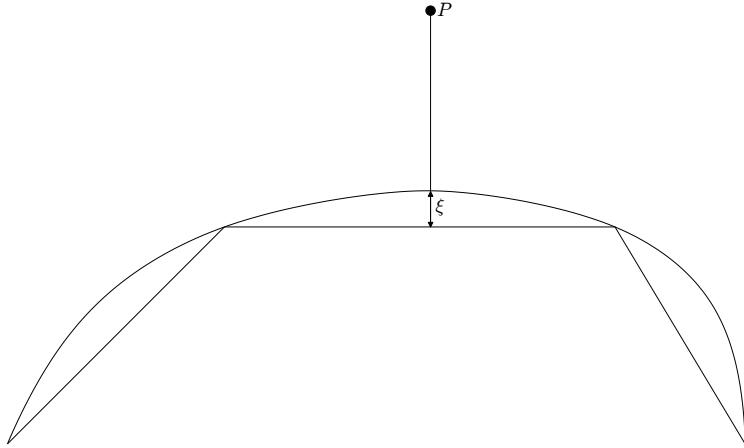
Ainsi, pour calculer le champ distance, la méthode du *Fast Sweeping* développée par Zhao (2005) est apparue comme idéale pour résoudre l'équation Eikonal dans le cas du calcul d'un champ distance, et c'est cette méthode que nous avons implémentée.

## 2.6 Propagation d'erreur à partir des surfaces discréétisées

Dans cette section, nous allons démontrer l'importance de l'initialisation des conditions frontières lors du calcul du champ distance d'un solide. La discréétisation d'une surface paramétrique donne toujours lieu à une erreur de représentation. Nous allons montrer que cette erreur de représentation, même faible, est propagée à travers tout le champ distance lors de son calcul et que la précision du champ distance peut être améliorée en initialisant les conditions frontières directement à partir de la surface paramétrique.

### 2.6.1 Erreur de représentation des surfaces discrétisées

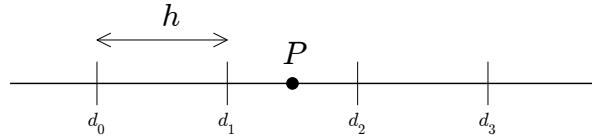
Comme le montre la figure 2.9, on introduit une erreur  $\epsilon$  lors de l'initialisation de nœuds frontières si on les initialise en effectuant une projection sur une discrétisation de la surface.



**Figure 2.9:** Erreur commise lors de la projection d'un point sur une géométrie discrétisée. Le point  $P$  est situé à la distance  $d$  de la géométrie originale, mais à la distance  $d + \epsilon$  lors de la projection sur la géométrie discrétisée.

### 2.6.2 Sensibilité aux conditions frontières

Il est important de comprendre l'influence des conditions frontières sur la résolution de l'équation de Eikonal. L'erreur commise par le schéma numérique du *Fast Sweeping* est bornée en  $O(|h \log h|)$ . Il est plus facile de comprendre l'effet de l'initialisation des points frontières sur la précision du résultat final en considérant le problème en une dimension. Le problème unidimensionnel consiste à déterminer la distance minimale d'un point à tous les nœuds d'un axe discrétisé (voir Fig. 2.10). On considère ainsi un axe discrétisé en plusieurs nœuds  $d_0, d_1, d_2$  et  $d_3$ . Chaque nœud est situé à la distance  $h$  de son voisin. On chercher à calculer la carte de distance du point  $P$  situé entre les nœuds  $d_1$  et  $d_2$  de l'axe. Le point  $P$  possède deux nœuds frontières, soit  $d_1$  et  $d_2$ . L'erreur  $\epsilon_1$  commise lors de l'initialisation du nœud  $d_1$  est propagée selon la caractéristique à droite et l'erreur  $\epsilon_2$  commise sur



**Figure 2.10:** Représentation de la propagation de l'erreur dans un cas unidimensionnel.

le noeud  $d_2$  est propagée à gauche. Il faut donc mesurer l'erreur selon la caractéristique de propagation. Contrairement à l'erreur du schéma, qui est bornée en  $O(|h \log h|)$ , l'erreur commise lors de l'initialisation des nœuds frontières est entièrement propagée jusqu'à l'extrémité du domaine. Ainsi, lors de l'initialisation, les nœuds frontières  $d_1$  et  $d_2$  ont pour valeur :

$$\begin{cases} \phi(d_1) = D_{d_1 \rightarrow P} + \epsilon_{d_1}, \\ \phi(d_2) = D_{d_2 \rightarrow P} + \epsilon_{d_2}. \end{cases}$$

Lors de la propagation de la distance aux nœuds  $d_0$  et  $d_3$ , on obtient les valeurs suivantes :

$$\begin{cases} \phi(d_0) = \phi(d_1) + h = D_{d_1 \rightarrow P} + h + \epsilon_{d_1}, \\ \phi(d_3) = \phi(d_2) + h = D_{d_2 \rightarrow P} + h + \epsilon_{d_2}. \end{cases}$$

L'erreur commise lors de l'initialisation des nœuds frontières  $d_1$  et  $d_2$  à été entièrement propagée aux autres nœuds. Bien que cet exemple puisse paraître trivial, le même phénomène se produit pour des géométries complexes.

## 2.7 Résumé

En résumé, il est crucial au niveau du calcul d'un champ distance que les conditions limites soient les plus précises possible. Contrairement aux schémas de balayage *Fast Sweeping*, les schémas d'avance de front *Fast Marching* et éléments finis nécessitent une discrétisation de la géométrie paramétrique. Parmi les méthodes d'approximations locales, le schéma de Godunov est un schéma *upwind* où seules les conditions frontières proches de la surface sont nécessaires. Comme la méthode de résolution de l'équation Eikonal développée par Zhao utilise un schéma

de balayage et une discrétisation de Godunov, c'est cette méthode que nous avons choisie pour calculer le champ distance d'un solide paramétrique.

## CHAPITRE 3

### PROJECTION ORTHOGONALE

La projection est une opération essentielle qui entre dans la résolution de nombreux problèmes géométriques. Ce chapitre vise à définir les concepts mathématiques reliés à la projection orthogonale d'un point sur une surface paramétrique.

Il est important de prévenir le lecteur que la projection sur une surface paramétrique est très complexe. Selon l'application nécessitant son usage, différentes méthodes peuvent être mise en œuvre et plusieurs résultats sont admissibles. Dans notre cas, nous nous intéressons à la projection dans le cadre bien précis du calcul d'un champ distance, et plus spécifiquement comme mécanisme d'initialisation des conditions frontières pour un schéma de propagation.

Lors du calcul du champ distance d'un solide, la projection est utilisée pour initialiser les conditions frontières. Elle permet de calculer la distance minimale entre la surface du solide et un point de l'espace situé à proximité de celle-ci.

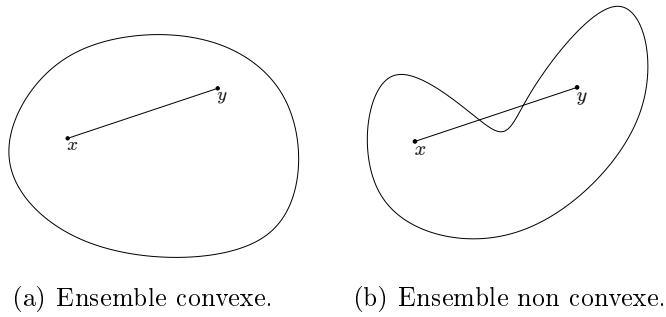
Dans un premier temps, nous rappellerons les principes mathématiques de la projection sur un convexe fermé ainsi que la méthode de base généralement utilisée pour calculer la projection d'un point sur une surface paramétrique. Dans un deuxième temps, nous effectuerons une revue bibliographique des différentes méthodes de projection plus élaborées développées récemment. Dans un troisième temps, nous évoquerons les difficultés inhérentes au calcul d'une projection sur une surface *NURBS*. Finalement nous proposerons une nouvelle approche de la projection basée sur l'optimisation d'une distance euclidienne.

#### 3.1 Projection orthogonale

Selon le théorème 3.1.2, qui généralise la projection orthogonale sur un espace vectoriel, la projection orthogonale sur un convexe revient à minimiser une distance (Luenberger, 1997). Rappelons brièvement la définition d'un ensemble convexe :

**Définition 3.1.1 (Ensemble convexe)** *Un ensemble  $C$  est dit convexe lorsque, pour tout  $x$  et  $y$  de  $C$ , le segment  $[x, y]$  est tout entier contenu dans  $C$ .*

La figure 3.1(a) représente un ensemble convexe, tandis que la figure 3.1(b) illustre un ensemble non convexe.



**Figure 3.1:** Convexité d'un ensemble.

On définit  $E$  comme un espace vectoriel muni d'un produit scalaire,  $x$  désigne un vecteur et  $C$  un ensemble convexe de l'espace complet  $E$ . La distance entre  $x$  et  $C$  désigne le minimum des distances entre  $x$  et un point de  $C$ . Le théorème de la projection s'énonce de la façon suivante :

**Théorème 3.1.2 (Projection sur un convexe fermé)** *Il existe une unique application  $t$  de  $E$  dans  $C$ , dite projection sur le convexe, qui à  $x$  associe le point  $t(x)$  de  $C$ , tel que la distance de  $x$  à  $C$  soit égale à celle de  $x$  à  $t(x)$ . Le vecteur  $t(x)$  est l'unique point de  $C$  vérifiant les deux propositions suivantes, qui sont équivalentes :*

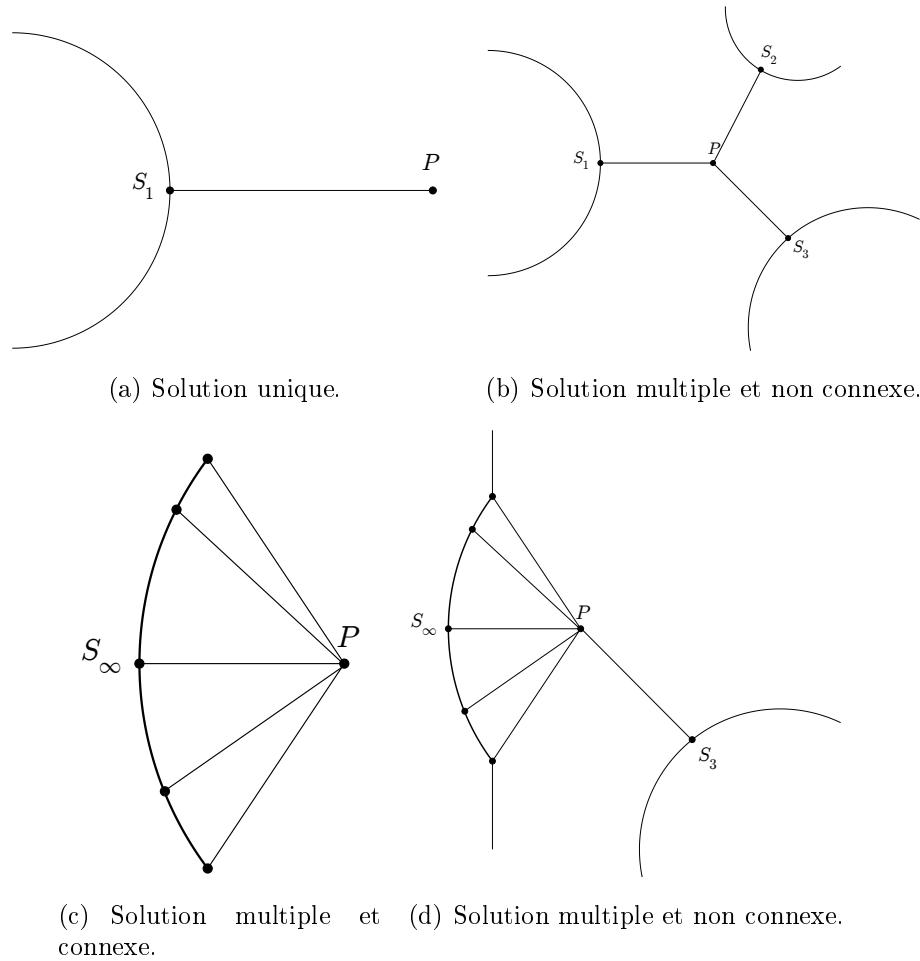
$$\forall y \in C, \quad \|x - t(x)\| \leq \|x - y\|, \quad (3.1)$$

$$\forall y \in C, \quad \langle x - t(x), y - t(x) \rangle \leq 0. \quad (3.2)$$

La projection orthogonale permet de déterminer la plus courte distance entre un point d'une part et une droite, un plan, ou plus généralement un sous-espace affine d'un espace euclidien d'autre part. À condition de travailler dans un espace euclidien (ou plus généralement dans un espace de Hilbert), on dispose d'un résultat

remarquable : étant donné un convexe fermé, pour tout point  $x$  de l'espace, il existe un et un seul point  $p(x)$  du convexe à distance minimale de  $x$ .

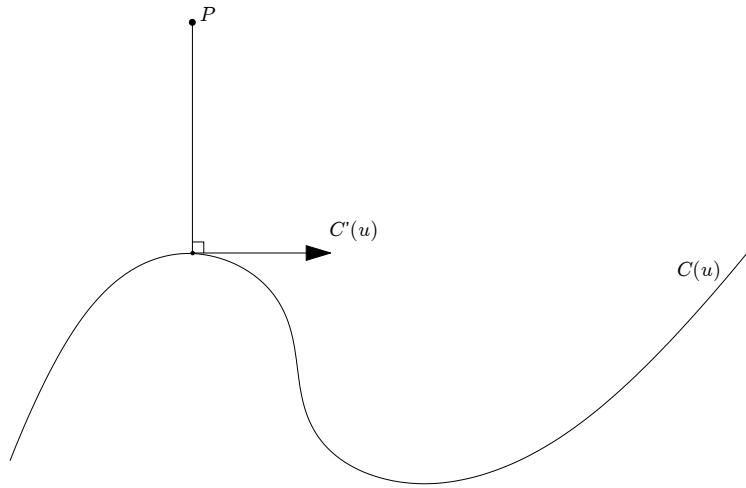
Dans le cas d'un ensemble non convexe, la projection est également obtenue en minimisant une distance euclidienne. Cependant, contrairement à la projection sur un ensemble convexe, il n'y a aucune garantie sur l'unicité de la solution. Celle-ci peut être unique, multiple et connexe, ou multiple et non connexe. Les problèmes relatifs à la projection sur un ensemble non convexe seront expliqués à la section 3.3.4.



**Figure 3.2:** Type de solution pour la projection.

### 3.2 Projection d'un point sur une surface paramétrique

Considérons la projection d'un point sur une courbe. Le calcul de la projection d'un point  $P$  sur une courbe paramétrique  $C(u)$ ,  $0 \leq u \leq 1$  consiste à déterminer la valeur optimale  $\bar{u}$  du paramètre  $u$  telle que le vecteur  $\overrightarrow{PC(\bar{u})}$  soit perpendiculaire à la tangente de la courbe au point  $C(\bar{u})$  (voir Fig. 3.2). On cherche donc la valeur



**Figure 3.3:** Projection orthogonale d'un point  $P$  sur une courbe paramétrique  $C(u)$ . Le vecteur  $\overrightarrow{PC(\bar{u})}$  est orthogonal à la tangente à la courbe au point  $C(\bar{u})$ .

du paramètre  $u$  telle que le produit scalaire entre le vecteur  $\overrightarrow{PC(u)}$  et la tangente à la courbe  $C'(u)$  soit nul. Ainsi, on optimise la fonction objectif  $f$  suivante :

$$f(u) = \vec{C}'(u) \cdot (\vec{C}(u) - \vec{P}).$$

Le paramètre optimal  $\bar{u}$  tel que  $f(\bar{u}) = 0$  est calculé numériquement à l'aide de l'algorithme de Newton. La fonction itérante suivante est utilisée :

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)},$$

$$u_{i+1} = u_i - \frac{C'(u_i) \cdot (C(u_i) - P)}{C''(u_i) \cdot (C(u_i) - P) + |C'(u_i)|^2}.$$

Trois critères d'arrêt sont utilisés afin de vérifier la convergence de la solution :

**Critère sur la distance.** La variation de la distance entre le point à projeter et la courbe est évaluée à chaque itération. Si la variation est inférieure à une très petite valeur  $\epsilon_1$ , on considère alors que l'algorithme de Newton a convergé vers une solution :

$$|\vec{C}(u_i) - \vec{P}| \leq \epsilon_1.$$

**Critère sur l'angle.** Si la variation de l'angle entre le vecteur  $\overrightarrow{PC(u)}$  et la tangente à la courbe est inférieure à  $\epsilon_2$ , alors les deux vecteurs sont considérés orthogonaux.

$$\frac{|C'(u_i) \cdot (C(u_i) - P)|}{|C'(u_i)| |C(u_i) - P|} \leq \epsilon_2.$$

**Critère sur la correction.** La correction est considérée comme suffisamment petite si le pas effectué entre deux itérations est inférieur à  $\epsilon_1$ .

$$|(u_{i+1} - u_i)C'(u_i)| \leq \epsilon_1. \quad (3.3)$$

Le processus itératif est interrompu lorsque les trois critères sont satisfaits.

Aucune relation mathématique entre les critères d'arrêts n'est fournie dans la littérature.

### 3.2.1 Projection sur une surface

La projection d'un point sur une surface suit une démarche analogue à la projection sur une courbe. Cependant, la projection sur une surface implique la résolution d'un système d'équations non linéaires. On définit la fonction vectorielle  $\vec{r}(u, v)$ , qui exprime le vecteur reliant le point  $\vec{P}$  à projeter et un point de la surface  $\vec{S}$  :

$$\vec{r}(u, v) = \vec{S}(u, v) - \vec{P}.$$

On définit aussi les deux fonctions scalaires  $f(u, v)$  et  $g(u, v)$  :

$$\begin{cases} f(u, v) = \vec{r}(u, v) \cdot S_u(u, v) = 0, \\ g(u, v) = \vec{r}(u, v) \cdot S_v(u, v) = 0. \end{cases} \quad (3.4)$$

Le système (3.4) implique que le vecteur  $\vec{r}(u, v)$  soit simultanément perpendiculaire à la tangente dans la direction  $u$  et à la tangente dans la direction  $v$ .

Afin de résoudre le système non linéaire (3.4) on définit les matrices suivantes :

$$\begin{aligned}\delta_i &= \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} u_{i+1} - u_i \\ v_{i+1} - v_i \end{bmatrix}, \\ J_i &= \begin{bmatrix} f_u & f_v \\ g_u & g_v \end{bmatrix} = \begin{bmatrix} |\vec{S}_u|^2 + \vec{r} \cdot \vec{S}_{uu} & \vec{S}_u \cdot \vec{S}_v + \vec{r} \cdot \vec{S}_{uv} \\ \vec{S}_u \cdot \vec{S}_v + \vec{r} \cdot \vec{S}_{vu} & |\vec{S}_v|^2 + \vec{r} \cdot \vec{S}_{vv} \end{bmatrix}, \\ \kappa_i &= - \begin{bmatrix} f(u_i, v_i) \\ g(u_i, v_i) \end{bmatrix}.\end{aligned}$$

La matrice  $\delta_i$  représente la correction et  $J_i$  le Jacobien. À chaque itération, on résoud le système suivant :

$$J_i \delta_i = \kappa_i.$$

à partir de  $\delta_i$  on obtient :

$$\begin{aligned}u_{i+1} &= \Delta u + u_i, \\ v_{i+1} &= \Delta v + v_i.\end{aligned}$$

Les quatre critères d'arrêt suivants sont généralement utilisés pour vérifier la convergence (Piegl et Tiller, 1995) :

**Critère sur la correction.** La correction est suffisamment petite,

$$|(u_{i+1} - u_i)S_u(u_i, v_i) + (v_{i+1} - v_i)S_v(u_i, v_i)| \leq \epsilon_1. \quad (3.5)$$

**Critère sur la distance.** La distance entre la surface et le point est très faible,

$$|S(u_i, v_i) - P| < \epsilon_2. \quad (3.6)$$

**Critère sur les cosinus.** Le cosinus de l'angle entre le vecteur  $\vec{r}$  et la tangente

est très proche de zéro,

$$\frac{|S_u(u_i, v_i) \cdot (S(u_i, v_i) - P)|}{|S_u(u_i, v_i)| |S(u_i, v_i) - P|} \leq \epsilon_3, \quad (3.7)$$

$$\frac{|S_v(u_i, v_i) \cdot (S(u_i, v_i) - P)|}{|S_v(u_i, v_i)| |S(u_i, v_i) - P|} \leq \epsilon_3. \quad (3.8)$$

Si la projection est située sur un bord d'une surface, alors un seul cosinus vérifie le critère d'angle.

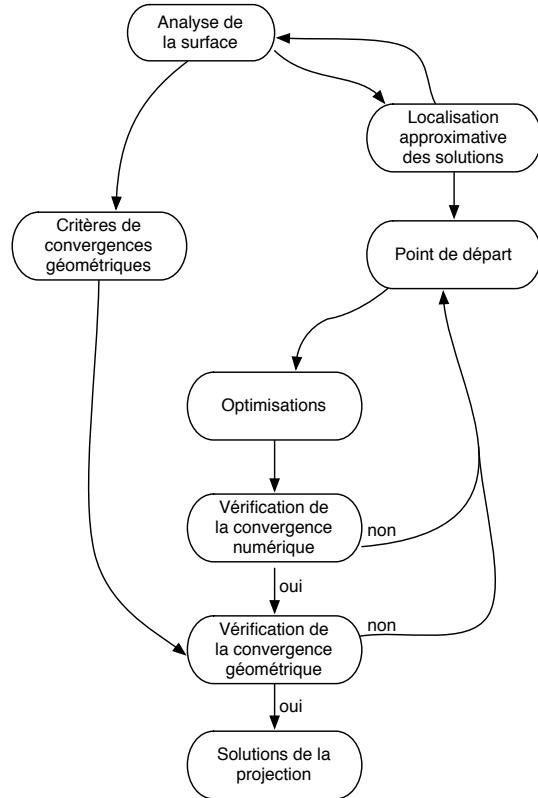
À chaque itération de Newton, il faut s'assurer que les paramètres restent à l'intérieur de l'intervalle de validité  $u, v \in [0, 1]$ .

### 3.2.2 Revue des méthodes de projection avancées

Il existe très peu de travaux sur le problème de la projection sur les modèles *B-Rep* paramétriques. La projection sur une surface se déroule en trois étapes. La première étape consiste à déterminer la localisation approximative de la solution. Le but de cette étape est de trouver un bon point de départ pour l'optimisation. La deuxième étape est l'optimisation. Cette étape optimise une fonction à partir du point de départ pour déterminer le projeté du point. La troisième étape est la vérification de la convergence de la solution. La figure 3.4 représente le cheminement général d'une méthode de projection.

Une des difficultés inhérentes au développement d'une méthode de projection est la localisation approximative de la solution afin de déterminer un point de départ adéquat pour l'optimisation.

Selimovic (2006) propose une méthode de projection sur une surface *NURBS* utilisant une subdivision successive de la surface. À chaque itération, la surface est subdivisée en deux sous-surfaces. En analysant la relation entre le point à projeter et le polygone de contrôle, une des deux sous-surfaces est rejetée. Ainsi, à chaque itération, on se rapproche de la solution recherchée en éliminant les sections de la surface qui ne contiennent pas la solution. Selimovic a proposé un théorème permettant de vérifier que le point de la surface  $S$  le plus proche d'un point  $P$  de



**Figure 3.4:** Étapes pour la projection.

l'espace est un des coins du polygone de contrôle. Ce théorème s'énonce comme suit :

**Théorème 3.2.1** *Soit  $S$  une surface NURBS et  $P$  le point à projeter sur la surface  $S$ .  $P_{i,j}$  est un nœud du polygone de contrôle et  $P_{0,0}$  est le nœud le plus proche de  $P$ . Si*

$$\langle P_{i,j} - P_{0,0}, P_{0,0} - P \rangle > 0,$$

*pour tout  $(i, j) \neq (0, 0)$ ,  $0 \leq i \leq m - p$  et  $0 \leq j \leq n - q$ , alors la distance minimale est*

$$\|S(u_0, v_0) - P\| = \text{dist}(P, S).$$

*Dans ce cas, tous les autres nœuds sont plus éloignés de  $P$  que  $P_{0,0}$ , et le point de la surface le plus proche de  $P$  est  $P_{0,0}$ .*

Selimovic utilise aussi un critère basé sur la définition d'un cône tangent. Cepen-

dant, il ne fournit aucun détail permettant de définir le cône tangent. La surface  $S$  est subdivisée jusqu'à l'obtention d'une surface planaire selon une tolérance donnée. Là encore, aucune valeur n'est précisée pour la tolérance. Une fois la surface *suffisamment* plane selon Selimovic, une méthode de Newton est alors mise en œuvre pour déterminer la projection orthogonale. Les équations utilisées pour la méthode de Newton ne sont pas données, mais on peut supposer que c'est la méthode classique proposée par Piegl et Tiller (1995) et qui a été présentée au début de ce chapitre.

Hu et Wallner (2005) proposent une méthode de projection d'ordre deux utilisant une approche géométrique et itérative. Cette méthode analyse la courbure normale de la surface. Un point de départ est choisi, puis, à l'aide des dérivées partielles et de la courbure normale, on calcule une correction pour obtenir un nouveau point. On procède ainsi de suite, de façon itérative, jusqu'à l'obtention approximative de la projection orthogonale.

Pour conclure cette revue des méthodes de projection, mentionnons que Ma et Hewitt (2003); Ma *et al.* (2006) ont aussi proposé une méthode de projection mais Chen *et al.* (2007) ont démontré qu'elle était erronée.

Toutes les méthodes de projections que nous venons de présenter sont basées sur la recherche d'une orthogonalité optimale. Toutefois, comme nous allons l'expliquer dans la suite de ce chapitre, cette approche est peut robuste.

### 3.2.3 Problèmes de robustesse existants

Dans certains cas de figure, la tangente et la normale d'une surface peuvent être dégénérées. Il n'existe aucune garantie de l'existence de la normale d'une surface, ce qui rend les méthodes de projection utilisant l'orthogonalité peut robuste.

### 3.2.3.1 Normales dégénérées

Le vecteur normal  $\vec{N}(u, v)$  d'une surface  $S(u, v)$  est défini comme le produit mixte des deux vecteurs tangents donnés par les dérivées partielles  $\vec{S}_u(u, v)$  et  $\vec{S}_v(u, v)$  :

$$\vec{N}(u, v) = \vec{S}_u(u, v) \times \vec{S}_v(u, v). \quad (3.9)$$

Le vecteur normal d'une surface peut être dégénéré, c'est-à-dire être égal au vecteur nul. Une normale dégénérée peut provoquer une erreur dans un algorithme. Afin d'éviter de telles erreurs, les dégénérescences doivent être détectées, et il est souhaitable d'utiliser une autre approche que l'équation (3.9) pour calculer la normale à la surface (Kiciak, 2001).

Selon l'équation (3.9), il est facile de voir que la normale est nulle si et seulement si un des deux cas suivants survient :

1. L'une des deux dérivées partielles ou les deux sont nulles :

$$S_u(u, v) = 0,$$

$$S_v(u, v) = 0.$$

2. Les dérivées partielles sont parallèles :

$$S_u(u, v) = \lambda S_v(u, v), \quad \lambda \neq 0.$$

Une analyse du polygone de contrôle peut, dans certains cas, indiquer la présence de normales dégénérées sur les bords de la surface. Une surface dont les côtés dégénèrent en une singularité possède une normale dégénérée en ce point. C'est le cas, par exemple, d'une sphère dont les nœuds du polygone de contrôle d'un côté sont confondus en un point pour former un pôle.

Les études Farin (1993); Seoul (1995); Kim *et al.* (1995) montrent qu'il est difficile de localiser les normales dégénérées en analysant simplement le polygone de contrôle de la surface. En effet, le vecteur normal peut être nul en n'importe quel point de la surface, aussi bien à l'intérieur que sur un bord ou un coin. Les tangentes peuvent

aussi devenir nulles lorsqu'un nœud est inséré plusieurs fois dans le vecteur nodal.

Aumann *et al.* (1995) donnent d'autres exemples de surfaces ayant une normale dégénérée à l'intérieur de la surface due à des dérivées partielles nulles. Kiciak (2001) a élaboré une méthode basée sur la construction d'une surface normale, permettant de calculer le vecteur normal en un point dégénéré.

### 3.3 Projection par optimisation d'une distance

Notre objectif est de déterminer le projeté d'un point sur un solide représenté par des surfaces *NURBS*. Une méthode de projection doit être robuste, c'est-à-dire, fournir un résultat valide dans tous les cas de figure. La précision du résultat est aussi primordiale. Nous désirons une méthode de projection nous permettant de contrôler l'erreur. La méthode proposée par Piegl et Tiller (1995) est basée sur l'orthogonalité de la projection. La plupart des travaux se concentrent sur la localisation approximative de la solution. Dans notre cas, c'est l'optimisation qui nous préoccupe.

Nous allons proposer une nouvelle approche basée sur la minimisation d'une distance. Cette approche devrait permettre de résoudre certains problèmes inhérents à l'approche orthogonale, tels que la projection sur un objet formé de plusieurs surfaces où la continuité n'est pas garantie au niveau des raccords. Les projections orthogonales sont basées sur la tangente. Cependant, comme nous l'avons déjà expliqué, rien ne garantit que la tangente d'une surface paramétrique est non nulle. Dans ce cas, une projection orthogonale ne fonctionne pas, contrairement à une projection par minimisation de la distance.

#### 3.3.1 Expression de la fonction donnant la projection

D'après le théorème 3.1.2, on peut déterminer la projection orthogonale en minimisant la distance d'un point de l'espace à la surface. On cherche donc à minimiser la fonction  $D(u, v)$ , qui représente le carré de la distance entre le point  $\vec{P_0}$  de l'espace

et un point d'une surface de Bézier :

$$D(u, v) = \left\| \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} (\vec{P}_0 - \vec{P}_{ij}) B_{i,n}(u) B_{j,m}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_{i,n}(u) B_{j,m}(v)} \right\|^2.$$

En posant  $\vec{V}_{ij} = \vec{P}_0 - \vec{P}_{ij}$ , on simplifie l'écriture de la distance au carré comme le produit scalaire du vecteur par lui même :  $\vec{V}_{ij} \cdot \vec{V}_{ij}$

$$D(u, v) = \sum_{k=1}^3 \left( \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} (\vec{V}_{ij})_k B_{i,n}(u) B_{j,m}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_{i,n}(u) B_{j,m}(v)} \right)^2.$$

Afin de minimiser la fonction  $D(u, v)$ , il est nécessaire de calculer son gradient ainsi que son Hessien :

$$\begin{aligned} \frac{\partial D(u, v)}{\partial u} &= \sum_{k=1}^3 \frac{\partial}{\partial u} \left( \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} (\vec{V}_{ij})_k B_{i,n}(u) B_{j,m}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_{i,n}(u) B_{j,m}(v)} \right)^2 \\ &= 2 \sum_{k=1}^3 \left( \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} (\vec{V}_{ij})_k B_{i,n}(u) B_{j,m}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_{i,n}(u) B_{j,m}(v)} \right) \\ &\quad \cdot \frac{\partial}{\partial u} \left( \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} (\vec{V}_{ij})_k B_{i,n}(u) B_{j,m}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_{i,n}(u) B_{j,m}(v)} \right). \end{aligned}$$

Pour alléger la notation on pose la fonction  $S_k(u, v)$  :

$$S_k = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} \left( \overrightarrow{V_{ij}} \right)_k B_{i,n}(u) B_{j,m}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_{i,n}(u) B_{j,m}(v)}.$$

On peut ainsi exprimer simplement la fonction  $D(u, v)$  :

$$D(u, v) = \sum_{k=1}^3 S_k^2.$$

La fonction  $S_k(u, v)$  est une surface de Bézier non rationnelle. On peut utiliser les algorithmes précédemment développés (Piegl et Tiller, 1995) pour calculer les dérivées. On obtient donc pour la variable  $u$  :

$$\frac{\partial D(u, v)}{\partial u} = 2 \cdot \sum_{k=1}^3 \frac{\partial}{\partial u} S_k \cdot S_k.$$

De façon similaire, pour la variable  $v$  :

$$\frac{\partial D(u, v)}{\partial v} = 2 \cdot \sum_{k=1}^3 \frac{\partial}{\partial v} S_k \cdot S_k.$$

Les dérivées partielles sont :

$$\frac{\partial D(u, v)}{\partial u \partial v} = 2 \sum_{k=1}^3 \frac{\partial}{\partial v} S_k \cdot \frac{\partial}{\partial u} S_k + S_k \frac{\partial}{\partial u \partial v} S_k.$$

La dérivée seconde en  $u$  est donnée par :

$$\begin{aligned}
 \frac{\partial^2 D(u, v)}{\partial u} &= \frac{\partial}{\partial u} \left( 2 \cdot \sum_{k=1}^3 \frac{\partial}{\partial u} S_k \cdot S_k \right) \\
 &= 2 \cdot \sum_{k=1}^3 \frac{\partial}{\partial u} S_k \cdot \frac{\partial}{\partial u} S_k + S_k \frac{\partial^2}{\partial u^2} S_k \\
 &= 2 \cdot \sum_{k=1}^3 \left( 2 \cdot \frac{\partial}{\partial u} S_k + S_k \frac{\partial^2}{\partial u^2} S_k \right) .
 \end{aligned}$$

De façon similaire, pour la variable  $v$  :

$$\frac{\partial^2 D(u, v)}{\partial v} = 2 \cdot \sum_{k=1}^3 \left( 2 \cdot \frac{\partial}{\partial v} S_k + S_k \frac{\partial^2}{\partial v^2} S_k \right) .$$

Comme nous connaissons l'expression analytique de la fonction  $D(u, v)$ , ainsi que ses dérivées première et seconde, nous sommes maintenant en mesure de la minimiser.

### 3.3.2 Minimisation de la distance

Nous cherchons à déterminer le minimum de la fonction  $D(u, v)$  sur le domaine  $u, v \in [0, 1]$ . La fonction  $D(u, v)$  est une fonction analytique dont nous pouvons évaluer le gradient et le Hessien. Nous avons donc utilisé une méthode de Newton pour minimiser cette fonction.

Nous avons utilisé la librairie *OPT++* (Meza *et al.*, 2007) qui permet d'optimiser une fonction multivariée soumise à des contraintes sur les bornes.

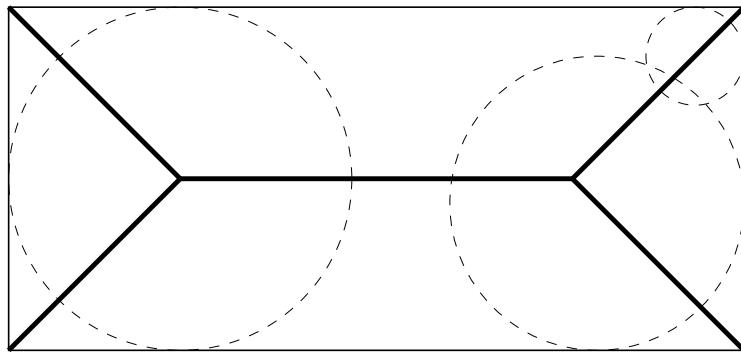
### 3.3.3 Choix du point de départ

Il est important de bien choisir le point de départ, car il influence le nombre d'itérations nécessaires à la convergence vers le minimum. Un bon point de départ permet à l'optimisation de converger vers un minimum global. On cherche donc un point de départ qui soit le plus proche possible de la solution.

Dans le cas de l'initialisation des conditions limites pour le calcul du champ distance, le point de départ sera déterminé selon la méthode expliquée à la section 4.5.2.1.

### 3.3.4 Solutions multiples au problème de la projection

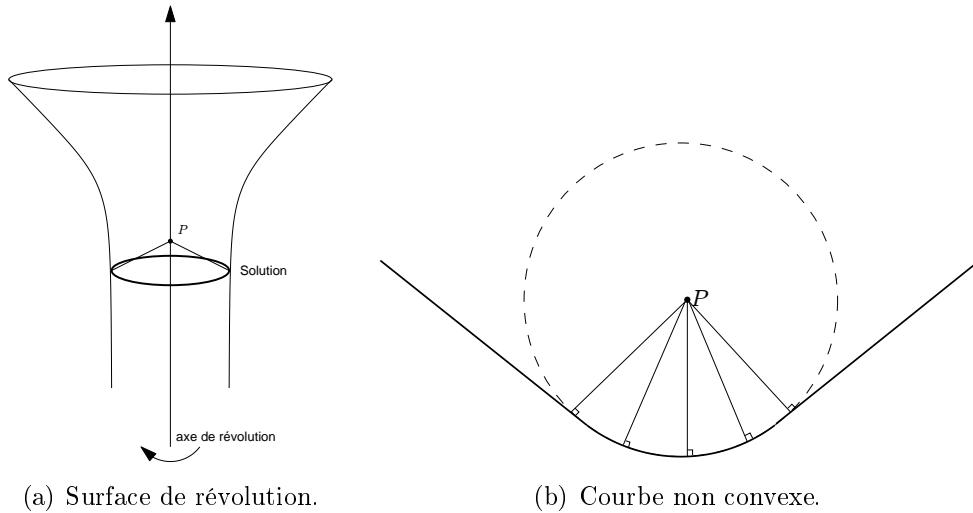
La géométrie peut présenter des configurations si complexes qu'il ne peut y avoir une seule et unique solution à la projection d'un point. Par exemple, la projection d'un point appartenant au squelette d'un solide n'admet pas une seule solution, car, par définition, un point du squelette est à égale distance d'au moins deux points non confondus de la surface. La figure 3.5 illustre l'axe médian d'un rectangle.



**Figure 3.5:** Axe médian d'un rectangle. L'axe médian est représenté en trait gras. Tous les points situés sur l'axe médian sont à équidistance d'au moins deux points distincts de la géométrie. Ainsi la projection d'un point appartenant à l'axe médian admet plusieurs solutions.

De façon similaire, la projection d'un point appartenant à l'axe de révolution d'une surface admet comme solution un cercle. La figure 3.6(a) illustre ce cas. Plus géné-

ralement, la projection de tout point situé au centre d'une courbure possède une infinité de solutions formées par un arc de cercle en 2D et une portion de sphère en 3D. La figure 3.6(b) illustre ce cas.

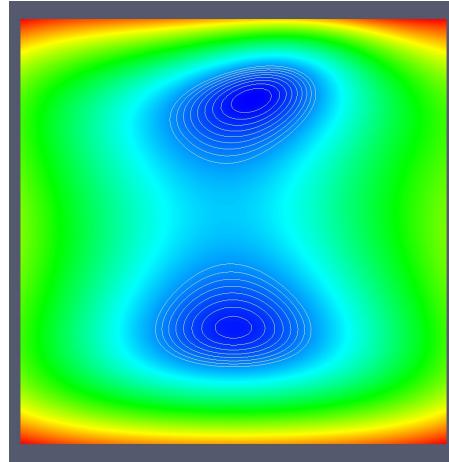


**Figure 3.6:** La projection peut admettre une infinité de solutions. Dans le cas de la projection d'un point de l'axe d'une surface de révolution, la solution forme alors un cercle. La projection du point situées au centre du cercle de rayon  $r$  admet une infinité de solutions située sur un arc de cercle.

### 3.3.5 Traitement des solutions multiples

Le problème de la projection peut posséder plusieurs solutions non connexes. Il y aura donc plusieurs minima à l'intérieur du domaine paramétrique. La figure 3.7 illustre un domaine paramétrique contenant deux minima.

Il n'existe pas de méthode publiée qui détermine toutes les solutions d'une projection. Dans le cas d'applications géométriques, il serait nécessaire de traiter les solutions multiples. Cependant, dans le cas qui nous occupe, c'est-à-dire le calcul de distances minimales, toutes les solutions sont à la même distance. Nous n'avons donc pas à nous soucier de la multiplicité des solutions de la projection.



**Figure 3.7:** Domaine paramétrique contenant deux minima.

### 3.3.5.1 Relations entre les critères de convergence

Afin de vérifier que la minimisation a bien convergé vers la solution, on vérifie l'orthogonalité dans la direction  $u$  et dans la direction  $v$  (Piegl et Tiller, 1995) :

$$\frac{|S_u(u, v) \cdot (S(u, v) - P_0)|}{|S_u(u, v)| |S(u, v) - P_0|} \leq \epsilon_1,$$

$$\frac{|S_v(u, v) \cdot (S(u, v) - P_0)|}{|S_v(u, v)| |S(u, v) - P_0|} \leq \epsilon_1.$$

## 3.4 Résumé

Les méthodes de projection sont complexes, car elles mettent en œuvre successivement des méthodes d'analyse de surface, de recherche géométrique et d'optimisation. Qui plus est, il est difficile d'unifier toutes ces méthodes et de déterminer des paramètres cohérents entre elles. Aucune méthode qui allierait une analyse de la surface et une évaluation robuste des tangentes lors de l'optimisation n'a encore été réalisée. Les critères de convergence sont fixés arbitrairement au lieu d'être reliés à une analyse de la surface. Les méthodes de projection actuelles se contentent de fournir une seule solution et ne traitent pas les solutions multiples.

Nous avons proposé une nouvelle approche à la projection en minimisant la distance plutôt qu'en optimisant l'orthogonalité comme il est d'usage. Cette formulation à l'avantage d'être bien adaptée au problème d'initialisation de points frontières et pose le problème de projection sous une forme qui peut être résolue de façon plus robuste du point de vue numérique.

La projection servira à initialiser les conditions limites pour un schéma de propagation. Elle servira aussi à localiser les noeuds formant les conditions frontières par rapport à la surface du solide.

## CHAPITRE 4

### CALCUL DU CHAMP DISTANCE D'UN SOLIDE PARAMÉTRIQUE

Ce chapitre décrit la méthodologie élaborée pour calculer le champ distance d'un solide représenté par des surfaces paramétriques. Nous effectuerons un bref rappel des objectifs à atteindre. Puis nous décrirons étape par étape la démarche élaborée afin de calculer la carte de distance d'un solide. Nous nous attarderons particulièrement sur l'initialisation des conditions frontières qui en est l'étape la plus difficile. Finalement nous présenterons une procédure robuste permettant d'orienter le champ distance d'un solide.

#### 4.1 Rappel des objectifs

Nous cherchons à calculer le champ distance d'un solide en préservant la continuité de la surface dans la solution et en évitant d'y ajouter inutilement des zones d'expansion. Pour obtenir une solution de continuité similaire à celle de la géométrie, nous voulons éviter de calculer le champ distance à partir d'un maillage surfacique de continuité  $C^0$  du solide, mais plutôt calculer la solution directement à partir de la surface originale de continuité,  $C^{p-1}$ .

Nous cherchons aussi à minimiser la propagation de l'erreur de représentation des conditions frontières lors de la résolution de l'équation Eikonal.

Étant donné qu'il est impossible de calculer analytiquement la carte de distance d'une surface quelconque, nous utiliserons une approche de résolution numérique nous permettant de contrôler l'erreur.

Dans ce projet, seul les solides représentés par un ensemble cohérent de surfaces paramétriques de type *NURBS* sont considérés.

## 4.2 Méthodologie

Le calcul du champ distance d'un solide peut être décomposé selon les cinq étapes suivantes :

1. Subdiviser la surface *NURBS* en surfaces de Bézier ;
2. Discréteriser le domaine de calcul à l'aide d'une grille cartésienne ;
3. Déterminer et initialiser les conditions frontières ;
4. Résoudre la formulation Eikonal de l'équation distance ;
5. Orienter le champ distance.

Chacune des étapes nécessaires au calcul du champ distance sera détaillée dans la suite de ce chapitre.

## 4.3 Subdivision de la surface du solide

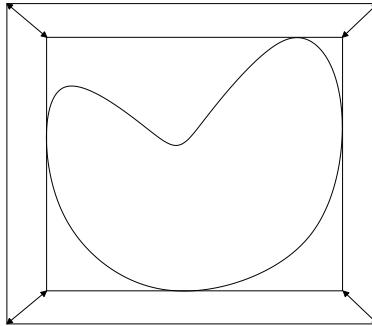
Une même surface B-spline peut représenter plusieurs changements de convexité, ce qui complique le calcul d'une projection, car la fonction optimisée présente alors plusieurs minima locaux. La décomposition du solide en surfaces élémentaires rend l'initialisation des conditions frontières plus facile et plus robuste en limitant le nombre de minimas locaux. Ainsi, afin de se rapprocher du cas idéal d'une surface convexe, on doit, dans une première étape, subdiviser la surface du solide en surfaces élémentaires moins complexes. On obtient alors un ensemble de surfaces géométriquement plus simples que la surface originale.

La surface *NURBS* originale est subdivisée en surfaces de Bézier rationnelles en insérant des nœuds dans le vecteur nodal. Les détails théoriques concernant la subdivision des surfaces *NURBS* sont développés en annexe II.4.

## 4.4 Définition et discréétisation du domaine de calcul

Les dimensions du domaine de calcul sont déterminées à l'aide de la boîte englobante du solide. Celle-ci est ajustée à l'aide d'un facteur d'expansion spécifié par

l'utilisateur afin de prévoir une région entourant le solide suffisante pour l'application considérée (voir Fig. 4.1).



**Figure 4.1:** Expansion du domaine de calcul.

Une fois les dimensions déterminées, le domaine de calcul est discréétisé à l'aide d'une grille cartésienne régulière. Une grille cartésienne régulière est une tessellation d'un espace euclidien en hexaèdres congruents, disposés de manière à remplir uniformément l'espace. Chaque hexaèdre forme une cellule. Les sommets des hexaèdres sont des *nœuds* auxquels sont attribués des indices uniques.

La méthode du *Fast Sweeping* garantit que l'erreur maximale  $\epsilon$  est de l'ordre de la taille de la grille  $h$  (Zhao, 2005) :

$$\epsilon \leq O(|h \log h|) .$$

Ainsi, la borne supérieure de l'erreur du champ distance dépend de la taille  $h$  de la grille. L'utilisateur doit spécifier une taille  $h$  de grille en fonction de la précision requise et des particularités importantes de la géométrie.

La méthode du *Fast Sweeping* ne nécessite pas la génération d'un maillage non structuré, mais simplement une grille cartésienne régulière. Il n'est ainsi pas nécessaire de discréétiser la géométrie pour raccrocher un maillage non structuré à la surface du solide. Par contre, l'utilisation d'une grille cartésienne complique la construction et l'initialisation des conditions frontières. En effet, les conditions frontières sont très faciles à déterminer sur un maillage non structuré, car elles sont formées par les nœuds du maillage surfacique et leur valeur initiale est nulle puisque

ces noeuds sont situés sur la surface.

## 4.5 Initialisation des conditions frontières

L'initialisation des conditions frontières est l'étape la plus importante et aussi la plus difficile à mettre en œuvre lors du calcul du champ distance d'un solide (Tsai, 2002). En effet, comme démontré à la section 2.6.2, l'exactitude des conditions frontières détermine la précision finale du champ distance. Les conditions frontières permettant de résoudre la fonction distance Eikonal sont constituées des noeuds de la grille cartésienne limitrophe à la surface du solide. La figure 4.3 illustre les conditions frontières d'un cercle.

L'initialisation des conditions frontières est réalisée en deux étapes. D'abord, il faut détecter les noeuds de la grille cartésienne qui sont limitrophes au solide. Ensuite, il faut initialiser ces noeuds avec la valeur exacte de  $\phi$ , c'est-à-dire la plus courte distance permettant de rejoindre la surface du solide.

### 4.5.1 Détection des noeuds frontières

Il convient dans un premier temps d'énoncer la définition d'un *noeud frontière* à une surface.

**Définition 4.5.1 (noeud frontière)** *Un noeud d'une grille cartésienne est un noeud frontière si un point de la surface est situé à l'intérieur d'une cellule à laquelle le noeud appartient.*

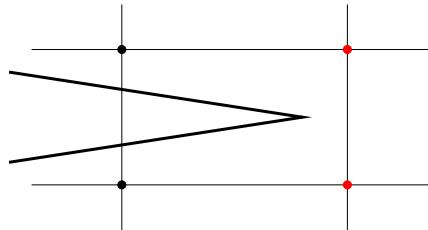
Afin de calculer et d'orienter le champ distance, il est nécessaire de déterminer les trois propriétés suivantes pour chaque noeud frontière :

1. La plus courte distance euclidienne permettant de rejoindre la surface du solide. Cette distance est nécessaire pour initialiser les conditions frontières.
2. Une projection du noeud sur la surface du solide. La projection permet de déterminer si le noeud est limitrophe au solide (voir théorème 4.5.3).

### 3. La localisation topologique du nœud, dans le cas d'un solide. La localisation permet d'orienter le champ distance.

Contrairement à ce qu'on pourrait penser, déterminer les nœuds d'une grille cartésienne qui sont limitrophes à la surface d'un solide n'est pas un problème trivial. Selon les propriétés connues des nœuds, plusieurs stratégies permettent de déterminer les nœuds frontières.

Une première méthode consiste à déterminer toutes les intersections entre tous les segments reliant les nœuds deux à deux et la géométrie (Wang *et al.*, 2005). Tous les sommets d'une cellule formant l'intersection sont considérés comme des nœuds frontières. Cette méthode dépend de la robustesse avec laquelle il est possible de déterminer l'intersection entre un segment et une surface (voir Fig. 4.2). Les problèmes liés à la robustesse sont difficiles et sont détaillés dans Edelsbrunner et Mücke (1990) et Shewchuk (1997).



**Figure 4.2:** Géométrie possédant un artefact aigu qui complique la détection des nœuds frontières. Le segment reliant les nœuds frontières situés à droite ne forment pas d'intersection avec la géométrie.

Une autre méthode consiste à classifier les nœuds de la grille selon leur orientation, c'est-à-dire ceux situés à l'intérieur ou à l'extérieur du solide. Si un nœud a un voisin d'orientation opposée, alors le nœud appartient à la frontière du solide. Cette méthode ne peut bien entendu fonctionner que pour des géométries fermées. Cependant, elle présente l'avantage que l'on travaille toujours dans le domaine continu. La localisation des nœuds ne peut se faire facilement que si l'on connaît une forme implicite de la géométrie. Or la transformation d'une géométrie paramétrique en une géométrie implicite est une opération ardue, ce n'est donc pas une approche évidente pour détecter les nœuds frontières.

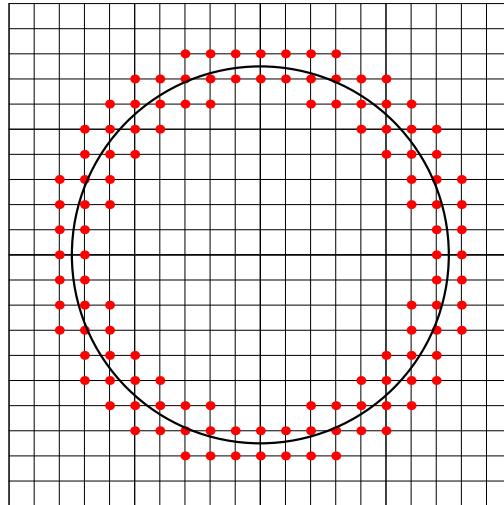
Une façon plus naturelle de déterminer si un nœud est limitrophe est de connaître la distance minimale de ce point au solide et de la comparer aux propriétés de la grille. Si la distance est plus petite que la diagonale de la grille, alors c'est un nœud frontière. On peut donc définir mathématiquement l'ensemble des nœuds de la grille qui sont limitrophes au solide.

**Définition 4.5.2** Soit  $E$  l'ensemble des nœuds d'une grille cartésienne et régulière de taille  $h$  et  $S$  l'ensemble des points formant la surface. La fonction  $D_{min}$  donne la distance minimale d'un point à la surface du solide. Alors, l'ensemble  $F$  des nœuds frontières est défini comme l'ensemble des nœuds dont la distance au solide est inférieure à la diagonale de la grille :

$$\left\{ F : E \mid D_{min}(E_i) \leq h\sqrt{2} \right\} \text{ en } 2D, \quad (4.1)$$

$$\left\{ F : E \mid D_{min}(E_i) \leq h\sqrt{3} \right\} \text{ en } 3D. \quad (4.2)$$

Cette définition n'est valide que pour une grille cartésienne régulière ; les cellules sont donc des carrés ou des cubes.



**Figure 4.3:** Nœuds frontières d'un cercle. La frontière est composée de tous les nœuds dont la distance au cercle est strictement inférieure à la diagonale de la grille.

Dans le cas général d'une grille cartésienne formée de rectangles ou de cuboïdes, la distance minimale ne suffit pas : il est indispensable de connaître la projection

du nœud sur le solide pour déterminer s'il est limitrophe. Le théorème suivant permet de généraliser la détection des nœuds frontières sur une grille cartésienne quelconque.

**Théorème 4.5.3 (Nœud frontière)** *Soit  $N$  un nœud d'une grille cartésienne structurée de dimension  $(h_x, h_y, h_z)$  et soit le point  $P$  la projection de  $N$  sur une surface. Soit  $\overrightarrow{NP}$  le vecteur reliant le nœud à sa projection sur la surface. Si  $|\overrightarrow{NP}_x| < h_x$  et  $|\overrightarrow{NP}_y| < h_y$  et  $|\overrightarrow{NP}_z| < h_z$ , alors le nœud  $N$  est un nœud frontière à la surface.*

Déterminer les nœuds frontières à l'aide de la distance nécessite une méthode de projection robuste et revient à calculer localement la valeur exacte du champ distance. Obtenir localement une solution exacte du champ distance nécessite une approche exhaustive. Il est donc important de limiter le nombre de points calculés de cette manière. On cherche donc à restreindre dans le voisinage immédiat de la surface le calcul exact des distances minimales.

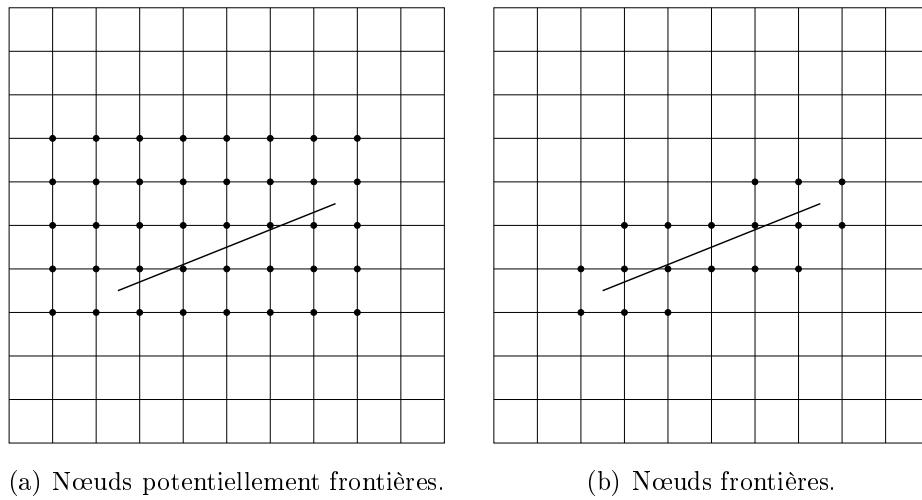
Nous nous sommes limité à l'utilisation d'une grille régulière. Les grilles rectangulaires ne sont pas traitées ici.

La détection des nœuds frontières est réalisée en deux étapes. D'abord, on détermine approximativement les nœuds les plus proches, ensuite, on élimine les nœuds encore trop éloignés en calculant la distance minimale exacte. Les nœuds restant forment alors les conditions frontières.

Afin de localiser approximativement les nœuds frontières d'une surface de Bézier, on effectue une tessellation en triangle de cette surface, puis on détermine les nœuds frontières de chaque triangle. On détermine la boîte englobante de chaque triangle ; pour une question de robustesse, celle-ci est agrandie de la diagonale de la grille dans toutes les directions cartésiennes. Ensuite, pour chacun des nœuds de la grille se trouvant à l'intérieur de la boîte englobante, on détermine la distance minimale au triangle. Cette stratégie repose sur notre capacité à projeter rapidement un point sur une facette triangulaire. Le calcul d'une projection robuste sur un triangle n'est pas trivial. Une méthode de projection sur un triangle est présentée en annexe I. Brièvement, cette méthode minimise une distance exprimée à l'aide d'une fonction

quadratique. L'optimisation de la distance est réalisée en déterminant analytiquement les paramètres rendant les dérivées nulles. La méthode est très performante au niveau du temps de calcul et la solution est exacte.

Les nœuds de la grille dont la distance minimale est inférieure à la diagonale de la grille sont marqués et initialisés comme des conditions frontières.



**Figure 4.4:** Conditions frontières d'un segment. La représentation (a) montre tous les nœuds de la grille qui font potentiellement partie de l'ensemble des nœuds frontières au segment. La figure (b) représente les nœuds qui forment finalement les conditions frontières.

Dans le même ordre d'idées, une autre stratégie aurait été de subdiviser la surface de Bézier jusqu'à l'obtention de sous-surfaces suffisamment minces, puis de considérer tous les nœuds situés dans la boîte englobante de chaque sous-surface comme des nœuds frontières. Cependant, il est plus facile et plus rapide d'effectuer une tesselation en triangle de la surface.

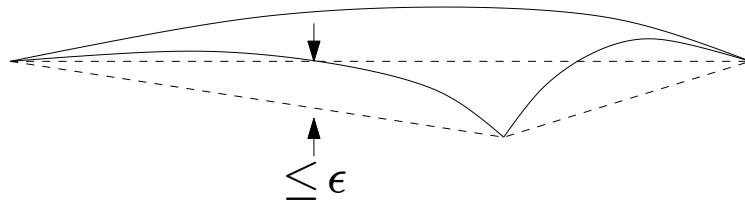
La précision des conditions frontières n'est en rien affectée par la tesselation en triangle, car les nœuds sont initialisés à partir de la surface originale. La tesselation de la surface ne sert qu'à localiser approximativement les nœuds les plus proches de la surface et à fournir un point de départ à la projection. La projection nous permet de déterminer la distance minimale entre le nœud frontière et la surface du solide avec le niveau de précision désiré. La projection est calculée selon la méthode développée au chapitre précédent.

Par ailleurs, on remarque que si l'on initialise plus de nœuds frontières que nécessaire, cela ne nuit pas à la précision, pourvu que leurs valeurs soient exacts. Ainsi, l'initialisation des conditions frontières est un compromis entre le temps nécessaire pour déterminer les nœuds frontières et une méthode exhaustive qui reviendrait à calculer localement le champ distance.

#### 4.5.1.1 Évaluer la profondeur de tessellation

Il est important de s'assurer que la surface est bien représentée par la tessellation en triangle. Nous avons utilisé la méthode proposée par Cheng (1992) afin de déterminer la profondeur de subdivision garantissant que l'erreur de représentation  $\epsilon$  maximale entre tous les triangles et une surface de Bézier est bornée. En effet, nous souhaitons que la distance maximale entre un triangle et la surface soit inférieure à la taille  $h$  de la grille.

La figure 4.5 illustre un triangle d'une tessellation ainsi que l'erreur de représentation commise. Une subdivision adaptative qui borne l'erreur de représentation



**Figure 4.5:** Représentation de l'erreur maximale  $\epsilon$  entre une surface et un triangle obtenu suite à la subdivision.

serait réalisable ; cependant une telle approche serait coûteuse en temps de calcul en raison de l'évaluation du critère d'erreur à chaque itération. Ainsi, l'approche développé dans Cheng (1992) consiste à déterminer dès le départ une profondeur de subdivision qui garantit la planéité (*flatness*), en effectuant une analyse du polygone de contrôle.

Dans un premier temps, définissons ce que nous entendons par la planéité  $\epsilon$ . Soit  $T$  un triangle dont les sommets  $v_i = (u_i, v_i)$ ,  $i = 0, 1, 2$  et soit  $h(u, v) : T \rightarrow E^3$

un triangle paramétrique de sommets  $S(v_i)$ ,  $i = 0, 1, 2$ . La triangulation  $S(T)$  de la surface  $S$  est dite *planaire* selon  $\epsilon$  si :

$$\sup_{(u,v) \in T} \inf_{(u',v') \in T} \|f(u,v) - h(u',v')\| \leq \epsilon.$$

La méthode de Cheng est très complexe et les démonstrations qui sous-tendent sa démarche dépassent le cadre de ce chapitre. Nous allons simplement donner les étapes des calculs nécessaires pour obtenir  $\alpha$ .

On considère une surface de Bézier rationnelle de degré  $n \times m$  et son polygone de contrôle formé par les points  $\vec{P}_{i,j}$  et leur pondération respective  $w_{i,j}$ . On désire connaître  $\alpha$ , la profondeur de subdivision permettant de garantir que l'erreur de représentation sera inférieure ou égale à  $\epsilon$ .

Les quatre étapes de calcul suivantes sont nécessaires pour déterminer la profondeur de subdivision  $\alpha$  :

1. Calculer les valeurs  $l$ ,  $r$  et  $L$  :

$$l = \min \{w_{i,j} \mid 0 \leq i \leq m, 0 \leq j \leq n\} ,$$

$$r = \max \left\{ \|\vec{P}_{i,j}\| \mid 0 \leq i \leq m, 0 \leq j \leq n \right\} ,$$

$$L = rl ;$$

2. Déterminer la valeur de  $\delta$  :

$$\delta = \begin{cases} \frac{l\epsilon}{(1 + (r - \epsilon)^2)^{1/2}}, & 0 < \epsilon \leq r, \\ l\epsilon, & r < \epsilon \leq 2r, \\ +\infty, & \epsilon > 2r; \end{cases}$$

3. Déterminer les valeurs  $M_1$ ,  $M_2$ ,  $M_3$ , qui représentent les valeurs maximales des dérivées :

$$M_1 = \sup_{(u,v) \in [0,1]} \left\| \frac{\partial^2 S(u,v)}{\partial u^2} \right\| ,$$

$$M_1 = m(m-1) \max \|\vec{P}_{i+2,j}^w - 2\vec{P}_{i+1,j}^w + \vec{P}_{i,j}^w\|,$$

pour  $0 \leq i \leq m-2$  et  $0 \leq j \leq n$ .

$$M_2 = \sup_{(u,v) \in [0,1]} \left\| \frac{\partial^2 S(u,v)}{\partial u \partial v} \right\|,$$

$$M_2 = mn \max \|\vec{P}_{i+1,j+1}^w - \vec{P}_{i+1,j}^w - \vec{P}_{i,j+1}^w + \vec{P}_{i,j}^w\|,$$

pour  $0 \leq i \leq m-1$  et  $0 \leq j \leq n-1$ .

$$M_3 = \sup_{(u,v) \in [0,1]} \left\| \frac{\partial^2 S(u,v)}{\partial v^2} \right\|,$$

$$M_3 = n(n-1) \max \|\vec{P}_{i,j+2}^w - 2\vec{P}_{i,j+1}^w + \vec{P}_{i,j}^w\|,$$

pour  $0 \leq i \leq m$  et  $0 \leq j \leq n-2$ .

4. Déterminer la profondeur de subdivision  $\alpha$  :

$$\alpha = \begin{cases} 0, & \text{si } M_1 + 2M_2 + M_3 < 8\epsilon, \\ 0, & \text{si } \delta = +\infty, \\ \left\lceil \log_4 \left( \frac{1}{8\delta} (M_1 + 2M_2 + M_3) \right) \right\rceil, & \text{autrement.} \end{cases} \quad (4.3)$$

Une fois la profondeur de subdivision  $\alpha$  déterminée, on divise le domaine paramétrique en  $2^{\alpha+2}$  triangles.

#### 4.5.2 Initialisation des conditions frontières

Une fois les nœuds frontières de la grille détectés, il faut les initialiser afin de former les conditions frontières nécessaires à la résolution de l'équation Eikonal. L'initialisation consiste à déterminer la distance minimale exacte entre chacun des nœuds et la surface. La plus courte distance est donnée par la projection du nœud sur la surface du solide. La méthode de projection utilisée, et qui a été préalablement présentée au chapitre 3, nécessite cependant un point de départ.

#### 4.5.2.1 Point de départ pour la projection

Comme mentionné au chapitre précédent, le choix du point de départ est crucial pour obtenir une bonne convergence de la méthode d'optimisation.

On utilise la projection sur la surface triangulée comme point de départ pour la méthode de projection. En effet, la distance maximale entre le maillage surfacique et la surface paramétrique est considérée suffisamment faible pour obtenir une bonne approximation de la solution en utilisant la projection sur le maillage comme point de départ.

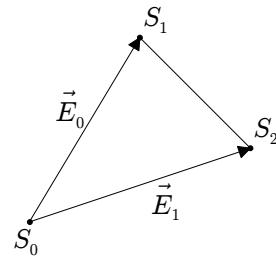
La projection d'un nœud sur une facette triangulaire est calculée en optimisant une distance euclidienne. Chaque triangle est exprimé sous forme la paramétrique suivante (voir Fig. 4.6) :

$$T(s, t) = S_0 + s\vec{E}_1 + t\vec{E}_2,$$

tel que :

$$T(s, t) \in D = \{(s, t) : s \in [0, 1], t \in [0, 1], s + t \leq 1\}.$$

Le calcul de la projection consiste à déterminer les paramètres  $(\bar{s}, \bar{t})$  optimaux tels



**Figure 4.6:** Triangle paramétrique.

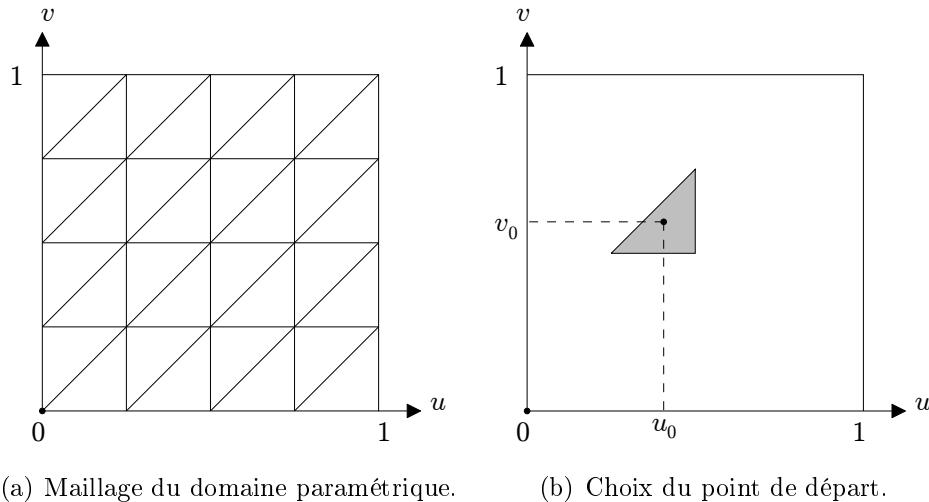
que la distance entre le nœud et la surface du triangle soit la plus courte possible. Les détails sont présentés en annexe I.

Une fois la projection sur le maillage surfacique obtenue, il faut déterminer un point de départ  $(u_0, v_0)$  situé sur la surface paramétrique. Le point  $T(\bar{s}, \bar{t})$  n'étant pas situé sur la surface  $S(u, v)$ , on utilise donc la transformation  $(\bar{s}, \bar{t}) \rightarrow (u_0, v_0)$

suivante pour ramener approximativement le point sur la surface :

$$\begin{cases} u_0 = S_{0u} + \bar{s}\vec{E}_{1u} + \bar{t}\vec{E}_{2u}, \\ v_0 = S_{0v} + \bar{s}\vec{E}_{1v} + \bar{t}\vec{E}_{2v}, \end{cases}$$

où  $S_0, S_1, S_2$  sont les sommets du triangle interpolé par la surface *NURBS*, et  $s, t$  les paramètres. Le calcul du point de départ est illustré à la figure 4.7(b).



**Figure 4.7:** Le domaine paramétrique est triangulé. Les paramètres  $(\bar{s}, \bar{t})$  issus de la projection sont transformés en point de départ  $(u_0, v_0)$ .

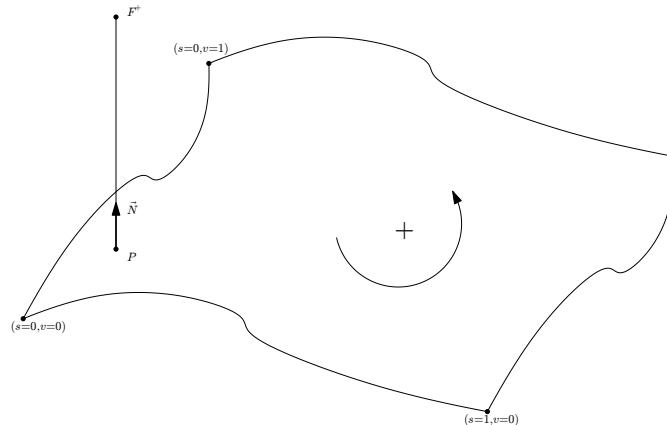
#### 4.5.2.2 Projection des nœuds frontières

Une fois un point de départ déterminé, on calcul la projection du nœud frontière sur la surface de Bézier à l'aide de la méthode développée au chapitre 3.

Bien qu'il puisse y avoir plusieurs solutions au problème de la projection, la distance est la même pour toutes les solutions. Nous n'avons donc pas à nous soucier de ce problème dans le cas particulier de l'initialisation des conditions frontières.

#### 4.5.3 Orientation géométrique des nœuds frontières

Afin d'orienter le champ distance, il faut d'abord déterminer la localisation des nœuds frontières par rapport à la surface. L'orientation d'un nœud frontière est déterminée après avoir calculé sa projection sur la surface. On connaît alors les paramètres  $(s^*, t^*)$  correspondant à la projection du nœud sur la surface  $S(u, v)$ . Le signe du produit scalaire entre la normale au point  $(s^*, t^*)$  et la projection donne l'orientation du nœud frontière (voir Fig. 4.8). Si le produit scalaire est positif, le nœud est à l'extérieur et si le produit scalaire est négatif, le nœud est à l'intérieur du solide.



**Figure 4.8:** Orientation géométrique d'un nœud frontière. Le nœud est situé à l'extérieur par rapport à l'orientation de la surface.

#### 4.5.4 Algorithme d'initialisation des conditions frontières

L'algorithme 4.9 permet de récapituler les étapes nécessaires à l'initialisation des conditions limites.

### 4.6 Calcul du champ distance

Une fois que les conditions frontières sont déterminées, le champ distance du solide est calculé en utilisant la méthode du *Fast-Sweeping* présentée au chapitre 2.

---

**Algorithme** :Initialisation des conditions frontières

---

**Données** :

- $\Gamma$  est l'ensemble des nœuds frontières ;
- $\phi$  est le champ distance de dimensions  $I, J, K$  ;
- $\theta$  est l'ensemble des orientations des nœuds frontières ;
- $S$  est une surface de Bézier ;
- $h$  est la taille de la grille cartésienne ;
- $D$  est la diagonale de la grille.

**Résultat** : Les noeuds frontières initialisés et orientés

**début**

```

 $\alpha \leftarrow \text{DéterminerSubdivision}(S, h) ;$ 
 $T \leftarrow \text{Trianguler}(S, \alpha) ;$ 
On détermine les nœuds proches de la triangulation;
pour chaque triangle  $\in T$  faire
   $boite \leftarrow \text{DéterminerBoiteEnglobante}(triangle) ;$ 
  pour tous noeud  $\in boite$  faire
     $p \leftarrow \text{Projection}(triangle, noeud) ;$ 
     $distance \leftarrow \text{CalculerDistanceEuclidienne}(p, noeud) ;$ 
    si  $distance < 2 \times D$  alors
       $\Gamma \leftarrow \text{Ajouter}(\Gamma, noeud)$ 
On détermine les nœuds frontières de  $S$ ;
pour chaque noeud  $\in \Gamma$  faire
   $d \leftarrow \text{CalculerPointDépart}(S, noeud) ;$ 
   $(u^*, v^*) \leftarrow \text{Projection}(S, noeud, d) ;$ 
   $distance \leftarrow \text{CalculerDistanceEuclidienne}(S(u^*, v^*), noeud) ;$ 
  si  $distance > D$  alors
     $\Gamma \leftarrow \text{Suprimer}(\Gamma, noeud)$ 
  sinon
    On initialise la conditions limite;
     $\phi(noeud) \leftarrow distance ;$ 
     $\theta(noeud) \leftarrow \text{DéterminerLocalisation}(S, u^*, v^*, noeud) ;$ 
fin

```

---

**Figure 4.9:** Initialisation des conditions frontières.

---

L'algorithme 4.10 présente un des huit balayages nécessaires pour obtenir le champ distance d'un solide.

---

**Algorithme :** Propagation des conditions frontières

**Données :**

- $\phi$  est le champ distance de dimensions  $I, J, K$  ;
- $\Gamma$  est l'ensemble des nœuds frontières.

**Résultat :** Le champ distance après un balayage dans une des huit directions.

**début**

```

pour  $x \leftarrow 2$  à  $I - 1$  faire
  pour  $y \leftarrow 2$  à  $J - 1$  faire
    pour  $z \leftarrow 2$  à  $K - 1$  faire
       $id \leftarrow z \times I \times J + y \times I + x$  ;
      si  $id \in \Gamma$  alors
        |  $\phi(i, j, k) \leftarrow \Gamma(id)$  ;
      sinon
        |  $a \leftarrow \min(\phi(i - 1, j, k), \phi(i + 1, j, k))$  ;
        |  $b \leftarrow \min(\phi(i, j - 1, k), \phi(i, j + 1, k))$  ;
        |  $c \leftarrow \min(\phi(i, j, k - 1), \phi(i, j, k + 1))$  ;
        |  $\phi(i, j, k) \leftarrow \frac{(a+b+c)+\sqrt{3h^2+2(ab+bc+ac-a^2-b^2-c^2)}}{3}$  ;
    
```

**fin**

---

**Figure 4.10:** Propagation des conditions frontières.

---

## 4.7 Orientation du champ distance

L'orientation de la carte de distance consiste à attribuer un signe positif ou négatif à chaque nœud de la grille cartésienne en fonction de sa localisation par rapport à la surface du solide.

L'orientation des nœuds frontières a été déterminée lors de l'étape de détection et d'initialisation des conditions frontières. Il s'agit maintenant de déterminer l'orientation de tous les autres nœuds de la grille cartésienne à partir de l'orientation des nœuds frontières.

Les nœuds frontières isolent les points intérieurs et extérieurs du solide. On distingue deux frontières, celle formée par les nœuds frontières intérieurs et celle formée par les nœuds frontières extérieurs. Ainsi, un nœud de la grille, qui n'est pas un nœud frontière au solide, ne peut avoir comme voisin que des nœuds intérieurs ou que des nœuds extérieurs. Par conséquent, il est possible de déterminer l'orientation d'un nœud de la grille ayant pour voisin un nœud frontière. Les deux nœuds possèdent alors la même orientation. De proche en proche, nous pouvons propager l'orientation des nœuds frontières. L'algorithme 4.11 permet de déterminer l'orientation des nœuds du champ distance.

---

**Algorithme :** Orientation du champ distance

**Données :**

- $\phi$  est le champ distance de dimensions  $I, J, K$  ;
- $\Gamma$  est l'ensemble des nœuds frontières ;
- $\theta$  est l'ensemble des orientations des nœuds frontières.

**Résultat :** Le champ distance orienté.

**début**

```

  orientation ← +1 ;
  pour  $x \leftarrow 0$  à  $I$  faire
    pour  $y \leftarrow 0$  à  $J$  faire
      pour  $z \leftarrow 0$  à  $K$  faire
         $id \leftarrow z \times I \times J + y \times I + x$  ;
        si  $id \in \Gamma$  alors
          orientation ←  $\theta(id)$ ;
           $\phi(id) \leftarrow |\phi(id)| \times orientation$  ;
  fin

```

---

**Figure 4.11:** Orientation du champ distance.

---

## 4.8 Résumé

Nous avons présenté une nouvelle méthode permettant de calculer le champ distance d'un solide paramétrique. Il a été nécessaire de développer une méthode de détection et d'initialisation des conditions frontières. Dans un premier temps, nous avons construit un maillage surfacique afin de déterminer approximativement la

localisation des nœuds frontières. Dans un deuxième temps, nous avons projeté les nœuds sur le solide afin de détecter les noeuds limitrophes à la surface. La projection permet aussi de déterminer le distance minimale ainsi que l'orientation des nœuds frontière. Une fois les conditions limites construites, nous avons calculé le champ distance à l'aide de la méthode du *Fast-Sweeping*. Nous avons également développé une méthode permettant de propager l'orientation des conditions frontières à l'ensemble des nœuds de la grille cartésienne.

## CHAPITRE 5

### ANALYSE ET VALIDATION DES RÉSULTATS

Ce chapitre est consacré à la présentation et à l'analyse des résultats obtenus. Différents modèles géométriques seront présentés afin de valider les étapes nécessaires au calcul du champ distance d'un solide. L'étape la plus importante étant l'initialisation des conditions frontières, leur validation fera principalement l'objet de ce chapitre.

Dans un premier temps, nous présenterons les différentes normes utilisées pour évaluer la qualité d'une solution obtenue. Dans un deuxième temps, nous présenterons un modèle paramétrique d'une sphère qui servira aux validations de chacune des différentes étapes du calcul du champ distance. L'implémentation du *Fast-Sweeping* sera validée en premier, puis nous vérifierons la qualité des conditions frontières pour le modèle de la sphère. Enfin nous vérifierons que l'erreur converge avec la taille de la grille, ainsi que la robustesse de l'orientation du champ distance.

Pour conclure, nous montrerons les résultats obtenus pour le champ distance du *Utah Tea-Pot* et d'une surface gauche.

#### 5.1 Mesure de l'erreur

Il convient, dans un premier temps, de définir les normes discrètes utilisées afin de mesurer l'erreur d'une solution. Trois mesures d'erreur ont été choisies, une absolue et deux relatives. L'erreur absolue  $\|e\|_\infty$  permet de mesurer l'erreur maximale :

$$\|e\|_\infty = \max \|\phi(i) - \phi_e(i)\|.$$

La valeur du champ distance au nœud  $i$  est  $\phi(i)$  et sa valeur exacte est  $\phi_e(i)$ . Cette mesure permettra de s'assurer que l'erreur maximale est inférieure à la borne

déterminée par Zhao :

$$\|e\|_\infty \leq O(|h \log h|). \quad (5.1)$$

Les deux normes discrètes  $\ell^1$  et  $\ell^2$  suivantes permettent de mesurer l'erreur relative :

$$\|e\|_{\ell^2} = \sqrt{\frac{\sum_{i=1}^{i=n} (\phi(i) - \phi_e(i))^2}{\sum_{i=1}^{i=n} \phi_e(i)^2}}, \quad (5.2)$$

$$\|e\|_{\ell^1} = \frac{1}{n} \sum_{i=1}^{i=n} \frac{\|\phi(i) - \phi_e(i)\|}{\|\phi_e(i)\|}. \quad (5.3)$$

## 5.2 Validation du *Fast-Sweeping*

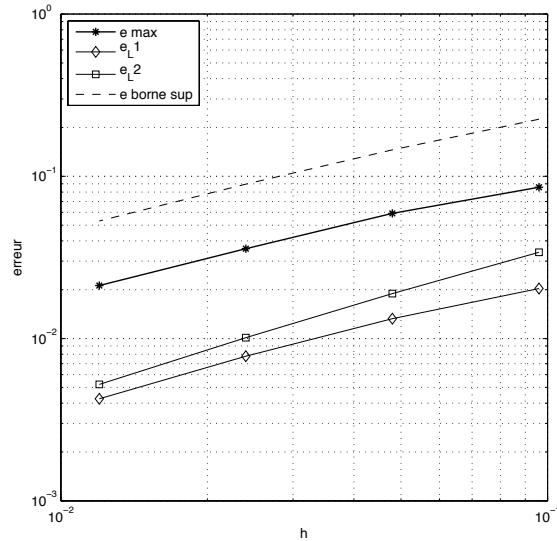
Dans un premier temps, il est important de valider notre implémentation du *Fast-Sweeping*. Les conditions frontières sont initialisées avec la distance exacte à la surface d'une sphère de rayon unitaire. Le champ distance est ensuite calculé à l'aide du *Fast-Sweeping* conformément à la méthode présentée au chapitre 2.

Les normes d'erreur  $\|e\|_\infty$ ,  $\|e\|_{\ell^1}$  et  $\|e\|_{\ell^2}$  sont évaluées selon les équations 5.2 et 5.3. Les valeurs des normes obtenues sont présentées dans le tableau 5.1 et sur le graphique 5.1.

**Tableau 5.1:** Convergence de l'erreur pour la méthode du Fast-Sweeping

$h$	0.096	0.048	0.024	0.012
$ h \log h $	0.2250	0.1458	0.0895	0.0531
$\ e\ _\infty$	0.0856459	0.059128	0.0358105	0.0212209
$\ e\ _{\ell^1}$	0.0203614	0.013251	0.0077900	0.0042521
$\ e\ _{\ell^2}$	0.0340242	0.018948	0.0101237	0.0052195

Comme on peut le constater sur le graphique 5.1, les normes  $\ell^1$  et  $\ell^2$  convergent linéairement avec la taille de la grille. L'erreur absolue  $\|e\|_\infty$  est toujours inférieure à la borne théorique  $O(|h \log h|)$ . Les résultats des normes d'erreurs sont comparables



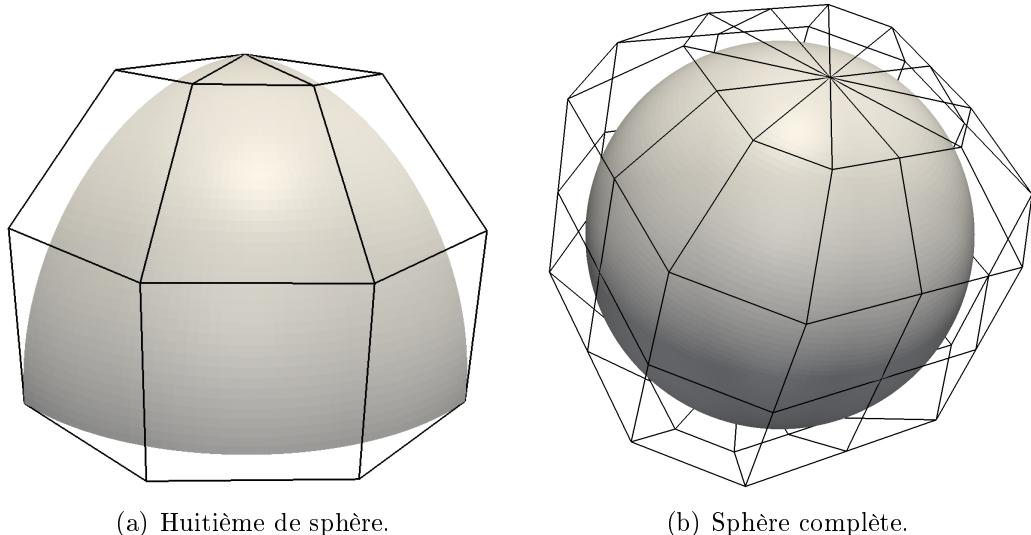
**Figure 5.1:** Graphique de convergence de l'erreur pour la méthode du *Fast-Sweeping*.

à ceux obtenus par Zhao (2005). Notre implémentation est donc valide et nous permet de calculer un champ distance. Nous sommes assuré de la convergence de l'erreur si les conditions frontières sont exactes.

### 5.3 Modèle bicubique d'une sphère

Le calcul exact du champ distance d'une géométrie quelconque est impossible. Afin de valider les schémas numériques développés, nous devons nous limiter à des formes géométriques simples dont le champ distance peut être calculé analytiquement. Un modèle paramétrique d'une sphère a été choisi pour les validations. La sphère est construite avec un assemblage de huit surfaces de Bézier bicubiques Grigor'ev (2008). Elle est représentée à la figure 5.2(b). Chacune des surfaces correspond à un huitième de sphère (voir Fig. 5.2(a)).

La continuité de la surface est  $C^3$ , sauf aux pôles où elle est  $C^1$  en raison de la singularité formée par les nœuds du polygone de contrôle qui dégénèrent en un point Yamaguchi (1997).



**Figure 5.2:** Polygone de contrôle d'une surface de Bézier bicubique représentant (a) un huitième de sphère et (b) une sphère complète.

## 5.4 Validation de champ distance d'un solide

Une fois notre implémentation du *Fast-Sweeping* validée, nous pouvons valider notre méthode de calcul du champ distance d'un solide paramétrique à l'aide du modèle de la sphère bicubique.

### 5.4.1 Validation des conditions frontières

Les conditions frontières sont formées par les nœuds de la grille les plus proches de la surface du solide. Pour chacun de ces nœuds, nous devons valider deux informations, soit la plus courte distance permettant de rejoindre la surface du solide et la localisation topologique du nœud par rapport au solide. Il est essentiel de valider les méthodes numériques, développées dans ce projet, permettant de déterminer ces deux informations. Bien entendu, il est aussi important de vérifier que tous les nœuds frontières ont bien été détectés et qu'il n'en manque aucun.

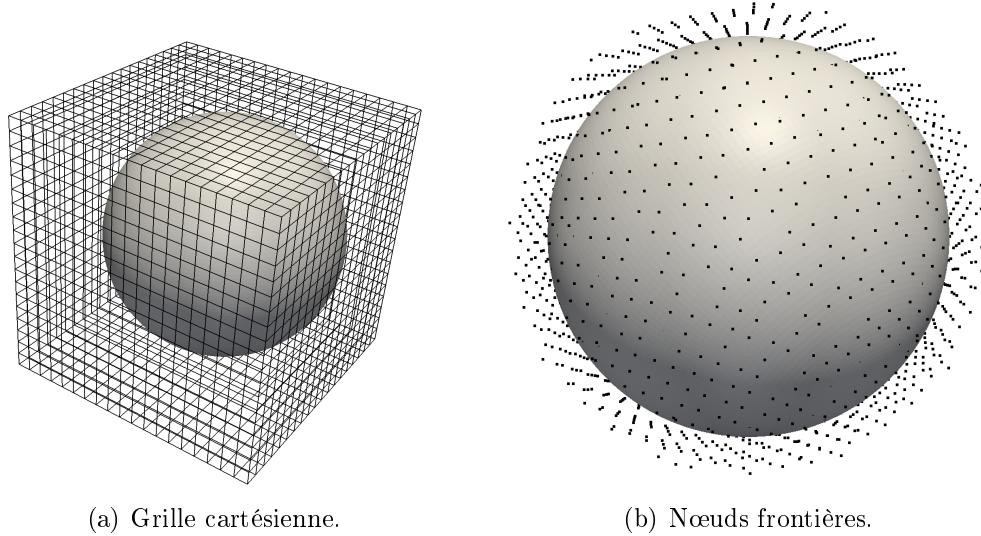
La fonction distance exacte pour une sphère de rayon unitaire centrée à l'origine

d'un repère est :

$$\phi(x, y, z)_{sphere} = \sqrt{x^2 + y^2 + z^2} - 1.$$

#### 5.4.1.1 Validation de la détection des nœuds frontières

Afin de valider la détection des nœuds frontières, nous avons vérifié que les nœuds de la grille situés à une distance inférieure à  $h\sqrt{2}$  de la surface font partie des conditions frontières. La figure 5.3(b) illustre l'ensemble des nœuds frontières d'une sphère.



**Figure 5.3:** Grille cartésienne et nœuds frontières d'une sphère.

Pour quatre grilles de discrétisation 0.12, 0.06, 0.03 et 0.015, tous les nœuds frontières ont été détectés. Il est donc raisonnable de penser que notre algorithme de détection des nœuds frontières est valide.

#### 5.4.1.2 Validation de l'initialisation des nœuds frontières

La projection des nœuds sur la surface est indépendante de la taille de la grille. L'erreur due à la projection n'est donc pas amenée à converger avec la grille. On ne calcule que la norme  $\|e\|_\infty$  pour les conditions frontières.

Le tableau 5.2 résume les résultats de l'analyse de l'erreur pour l'initialisation des conditions frontière.

**Tableau 5.2:** Analyse de l'erreur sur la distance des nœuds frontières pour le modèle de la sphère

$h$	0.12	0.06	0.03	0.015
$\ e\ _\infty$	0.00152762	0.00466901	0.00124248	0.00174968

L'erreur maximale est plutôt constante quelle que soit la taille de la grille.

#### 5.4.1.3 Validation de l'orientation des conditions frontières

L'orientation des conditions frontières a été validée en comparant le signe des conditions frontières exactes avec celui obtenu par la projection. Pour quatre grilles de discrétisation 0.12, 0.06, 0.03 et 0.015, tous les nœuds avaient la bonne orientation. Il est donc raisonnable de penser que notre algorithme permettant de déterminer la localisation topologique d'un nœud par rapport au solide est valide.

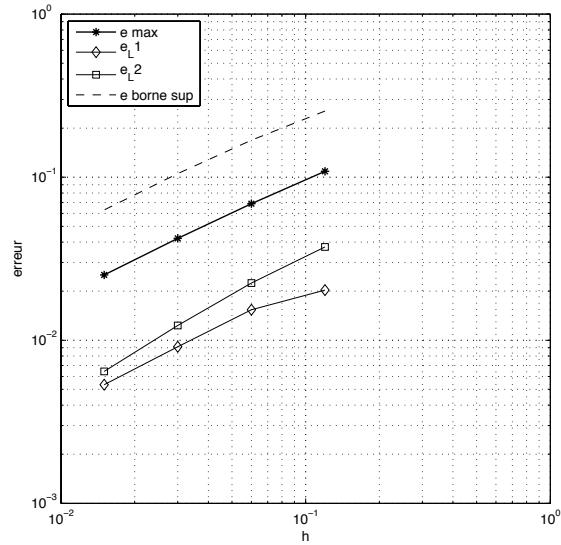
#### 5.4.2 Analyse de convergence de l'erreur du champ distance

Les normes  $\|e\|_\infty$ ,  $\|e_r\|_{\ell^1}$  et  $\|e_r\|_{\ell^2}$  ont été calculées pour l'ensemble des nœuds de la grille. Les résultats sont présentés dans le tableau 5.3. La figure 5.4 représente un graphique des différentes normes d'erreur en fonction de la taille de la grille.

**Tableau 5.3:** Tableau montrant que l'erreur maximale est toujours inférieure à  $|h \log h|$

$h$	0.12	0.06	0.03	0.015
$ h \log h $	0.2544	0.1688	0.1052	0.0630
$\ e\ _\infty$	0.10865	0.0688371	0.0421315	0.0251541
$\ e_r\ _{\ell^1}$	0.0201733	0.0151103	0.0091092	0.0052855
$\ e_r\ _{\ell^2}$	0.037344	0.0224376	0.0123222	0.0064447

On constate que toutes les normes d'erreur sont inférieures à  $|h \log h|$ . Les normes  $\|e\|_\infty$ ,  $\|e_r\|_{\ell^1}$  et  $\|e_r\|_{\ell^2}$  convergent linéairement avec la taille de la grille. Notre méthode de calcul du champ distance d'un solide paramétrique est donc valide.



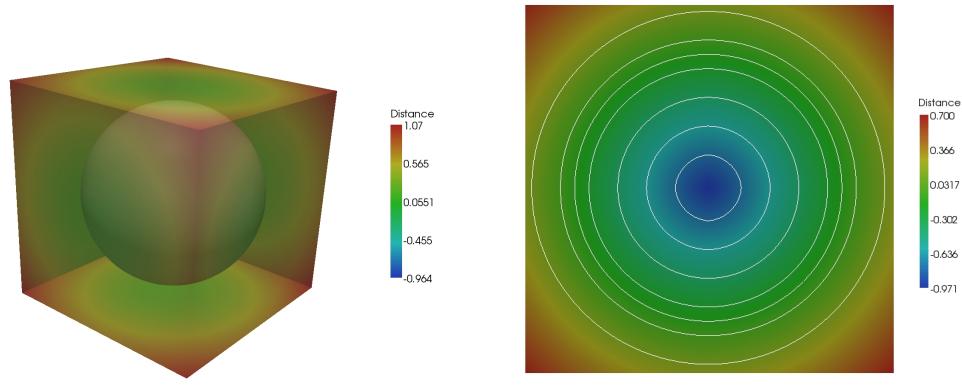
**Figure 5.4:** Graphique de l'erreur du champ distance.

#### 5.4.3 Validation de l'orientation du champ distance

L'orientation du champ distance a été validée en comparant l'orientation exacte avec celle propagée à partir des conditions frontières. Pour quatre grilles de discré-tisation 0.12, 0.06, 0.03 et 0.015, tous les nœuds avaient la bonne orientation. Notre algorithme permettant de propager la localisation topologique des nœuds est donc valide.

La figure 5.5(a) montre le champ distance orienté de la sphère.

Afin de montrer l'importance de l'initialisation de conditions limites, nous avons calculé le champ distance d'une sphère triangulée et nous l'avons comparé à celui d'une sphère paramétrique. La surface triangulée a été obtenue à l'aide d'une dis-crétisation de la surface paramétrique de manière à ce que, la distance de Hausdroff entre les deux surfaces soit inférieure ou égale à  $h$ . Les noeuds frontières ont été calculés avec une projection sur le maillage surfacique dans un cas et une projec-tion sur la surface paramétrique dans l'autre cas. Comme le montre la figure 5.6, une isosurface permet de comparer visuellement les deux champs distances. La sur-face isométrique provenant du champ distance de la sphère paramétrique (figure 5.6(b)) est beaucoup plus lisse que celle issue du champ distance de la même surface



(a) Champ distance d'une sphère. (b) Lignes iso d'une coupe du champ distance.

**Figure 5.5:** Champ distance d'une sphère.

triangulée (figure 5.6(a)), et ce pour une même grille de calcul du champ distance.

## 5.5 Exemples de champ distance

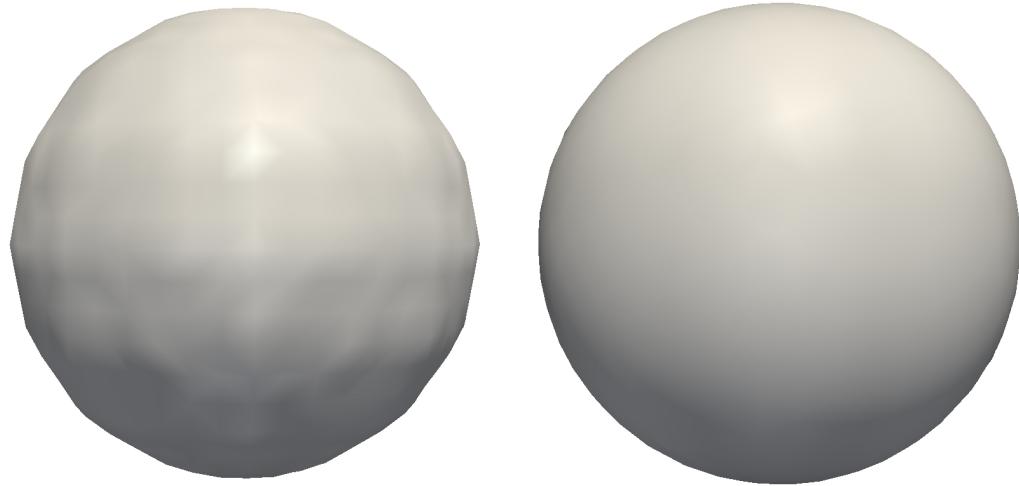
Nous allons présenter trois autres champs distance de solides paramétriques, un cylindre, un solide de révolution et le modèle bien connu du *Teapot*. Pour le cylindre et le solide de révolution, comme nous sommes en mesure de calculer leur champ distance exact, nous allons présenter une analyse de convergence de l'erreur.

### 5.5.1 Cylindre

La géométrie utilisée pour ce cas test est une surface cylindrique. Le solide complet est composé de douze surfaces de Bézier. La surface cylindrique est formée par quatre surfaces de Bézier représentant un quart de cylindre. Le cylindre est fermé par deux disques formés de quatre surfaces de Bézier chacun. Les surfaces sont orientées de manière à ce que les normales pointent vers l'extérieur du cylindre.

Les nœuds frontières du cylindre ont tous été détectés et sont illustrés à la figure 5.7(b).

Les normes discrètes  $\|e\|_\infty$ ,  $\|e_r\|_{\ell^1}$  et  $\|e_r\|_{\ell^2}$  ont été calculées pour l'ensemble des



(a) Isosurface pour le maillage surfacique. (b) Isosurface pour la surface paramétrique.

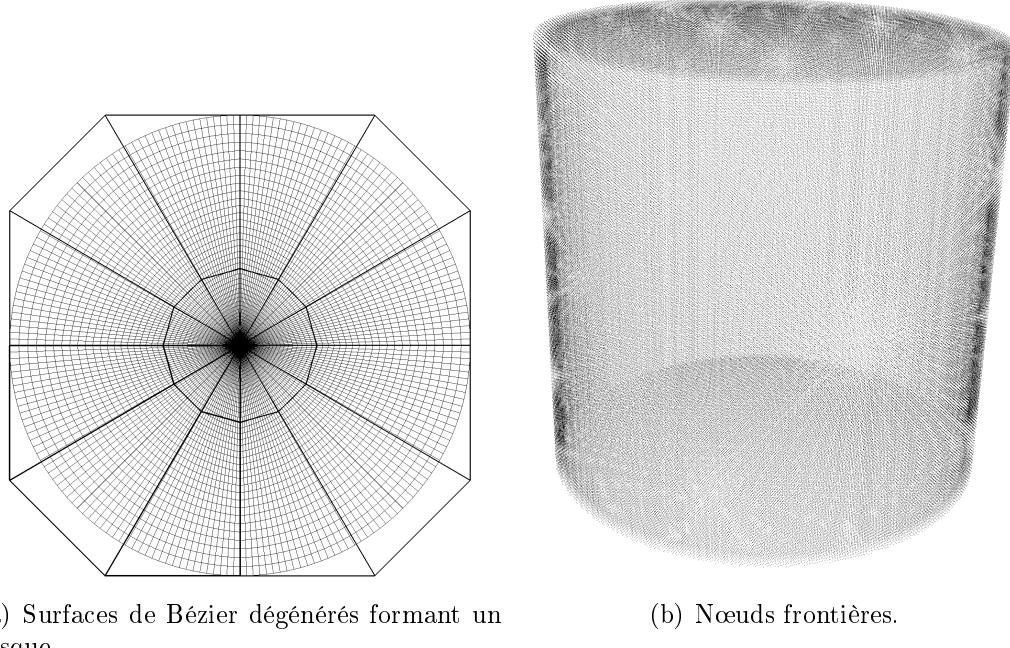
**Figure 5.6:** Comparaison entre l'isosurface pour un maillage surfacique tel que  $\epsilon < h$  et l'isosurface de la sphère paramétrique. En (a), les conditions frontières ont été construite à partir de la projection sur une discréétisation en triangle des surfaces de Bézier. En (b) les conditions frontières ont été construites à partir de la projection sur les surfaces de Bézier. On observe que l'isosurface de (b) est plus lisse que celle de (a).

nœuds du champ distance. Les résultats sont présentés dans le tableau 5.4 . Comme on peut le voir sur le graphique 5.8, toutes les normes d'erreur convergent avec  $h$ , et l'erreur maximale est toujours inférieure à  $|h \log h|$ .

Quatre nœuds frontières ont une mauvaise orientation. Ces nœuds sont situés au centre des disques du cylindre. Les surfaces de Bézier formant les disques sont particulièrement dégénérées, ce qui pose des problèmes lors des calculs des normales (voir Fig. 5.7(a)).

**Tableau 5.4:** Analyse de convergence de l'erreur pour le cylindre

$h$	0.12	0.06	0.03	0.015
$ h \log h $	0.2544	0.1688	0.1052	0.0630
$\ e\ _\infty$	0.0805048	0.0472304	0.0365004	0.017868
$\ e_r\ _{\ell^1}$	0.0137021	0.0113156	0.0075781	0.0046571
$\ e_r\ _{\ell^2}$	0.0367473	0.0223662	0.0136288	0.0082194

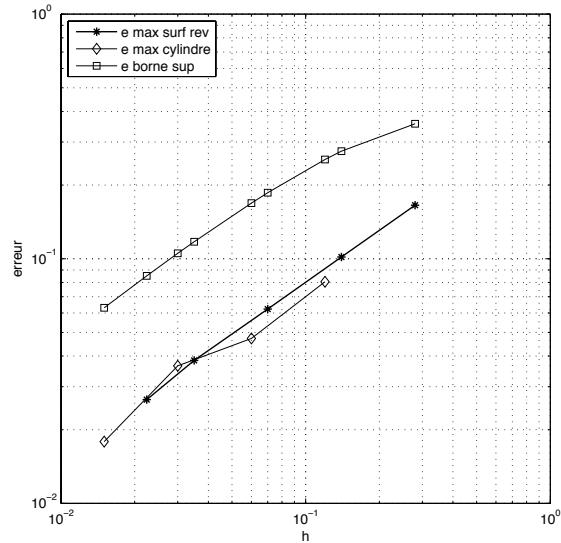


**Figure 5.7:** Disque bicubique dégénéré et nœuds frontières du cylindre.

### 5.5.2 Solide de révolution d'une parabole

La géométrie utilisée pour le deuxième test est un solide de révolution parabolique. La surface de révolution parabolique peut être représentée comme un demi-cercle de rayon unitaire dont l'extréma tourne autour de l'axe  $z$  selon un cercle lui aussi unitaire. Cette géométrie est illustrée à la figure 5.9(a). La géométrie complète est composée de douze surfaces de Bézier bicubiques. La surface de révolution est formée par quatre surfaces de Bézier (voir Fig. 5.9(b)). Ces surfaces sont surmontées de quatre autres surfaces formant un disque pour fermer la partie supérieure. La partie inférieure du solide est fermée par un autre disque. Les noeuds du polygone de contrôle sont ordonnés de manière à ce que les normales pointent vers l'extérieur. On obtient ainsi un solide de révolution paramétrique, fermé et orienté.

L'orientation des noeuds situés sur les bords du solide n'est pas robuste. En effet, les surfaces paraboliques sont confondues avec les disques supérieur et inférieur. Ainsi un nœud frontière de l'arête est à la fois un nœud frontière du disque et de la surface parabolique, or les deux normales sont de sens opposés à cet endroit. Il y



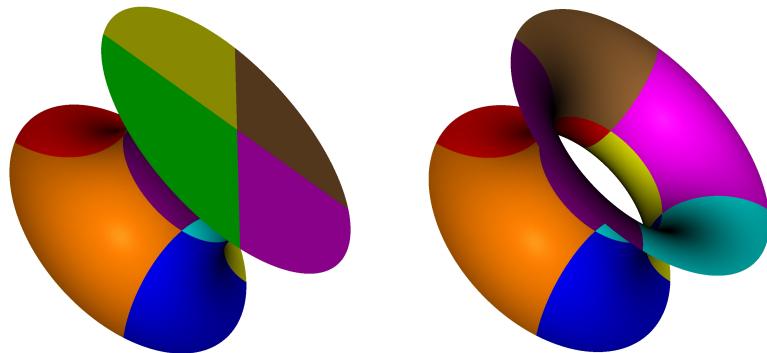
**Figure 5.8:** Erreur du champ distance du cylindre et de la surface de révolution.

a donc une incertitude sur la localisation du nœud par rapport à la surface. C'est un cas difficile à traiter pour des algorithmes géométriques.

Les normes  $\|e\|_\infty$ ,  $\|e_r\|_{\ell^1}$  et  $\|e_r\|_{\ell^2}$  ont été calculées pour l'ensemble des nœuds du champ distance. Les résultats sont présentés dans le tableau 5.5 et la figure 5.8 représente un graphique des différentes normes d'erreur en fonction de la taille de la grille. Les normes d'erreur convergent avec  $h$ , et l'erreur maximale est toujours inférieur à  $|h \log h|$ .

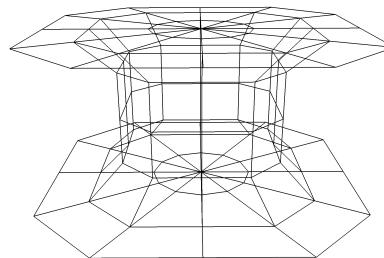
**Tableau 5.5:** Analyse de convergence de l'erreur pour le solide de révolution

$h$	0.28	0.14	0.07	0.035	0.0224
$ h \log h $	0.3564	0.2753	0.2497	0.1173	0.0851
$\ e\ _\infty$	0.165566	0.101615	0.0622818	0.0383766	0.0265511
$\ e_r\ _{\ell^1}$	0.0379004	0.034682	0.0259307	0.0183276	0.0129898
$\ e_r\ _{\ell^2}$	0.0699807	0.0501713	0.0346451	0.0225777	0.0158076

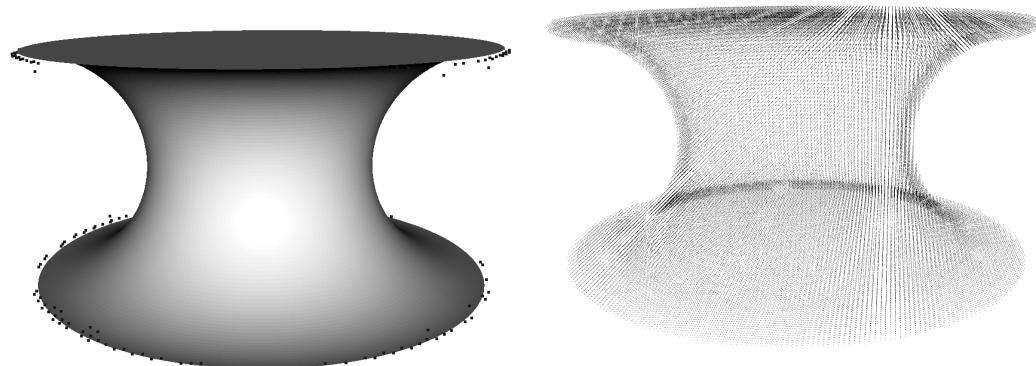


(a) Solide de révolution.

(b) Surface de révolution.



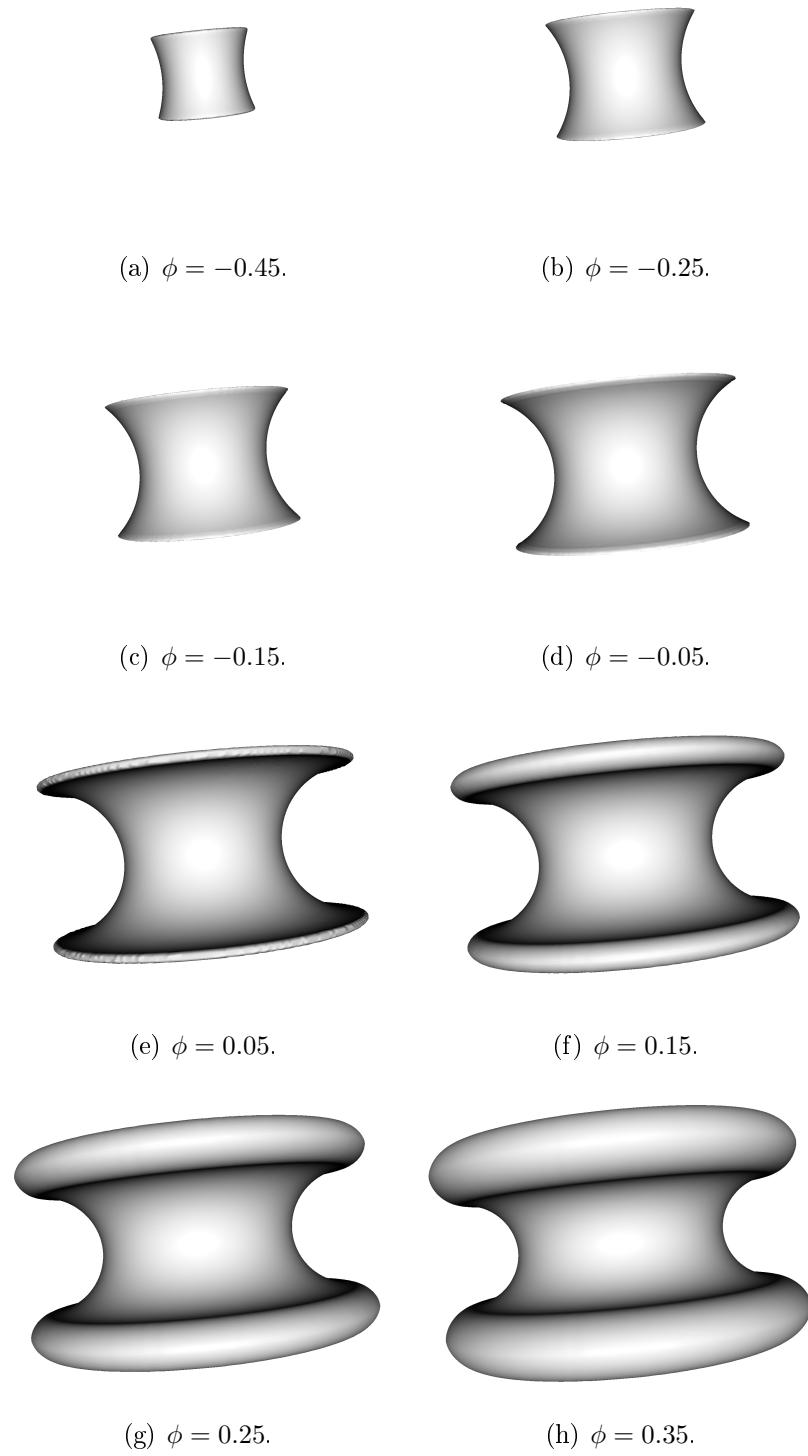
(c) Polygone de contrôle du solide de révolution.

**Figure 5.9:** Polygone de contrôle et géométrie du solide de révolution.

(a) Nœuds mal orientés.

(b) Nœuds frontières.

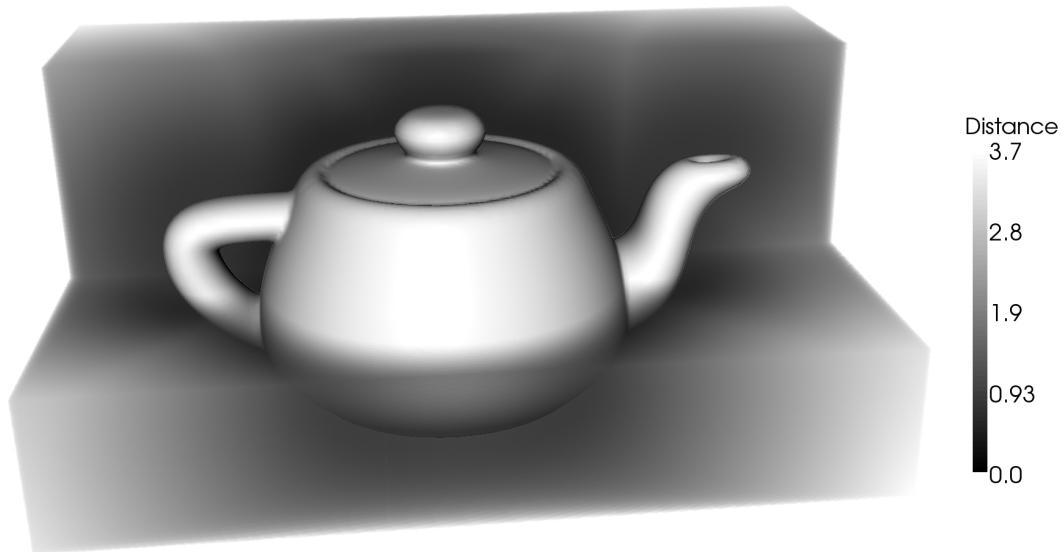
**Figure 5.10:** Nœuds frontières d'un solide de révolution et nœuds ayant une mauvaise orientation. Les nœuds ayant une mauvaise orientation sont situés sur les cercles formant la frontière entre les disques et les surfaces paraboliques.



**Figure 5.11:** Surfaces iso pour différentes valeurs  $\phi$  du champ distance d'une surface de révolution parabolique.

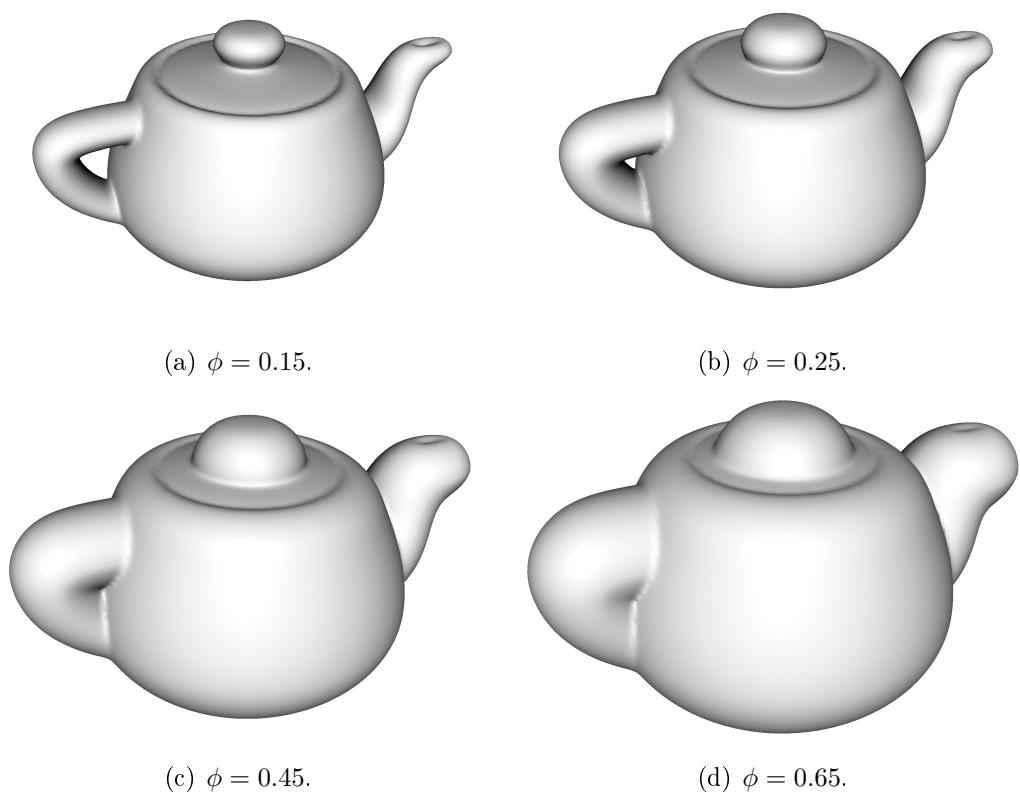
### 5.5.3 Le *Utah Teapot*

La théière de l'Utah (*Utah teapot*) est un modèle mathématique d'une théière. Sa géométrie est partiellement concave et contient des extrema. Son genre est supérieur à zéro à cause du trou dans la poignée. Cependant le modèle n'est pas fermé, on ne peut donc pas orienter son champ distance. La figure 5.12 et 5.13 illustrent différentes surfaces iso du champ distance du *Utah Teapot*. Nous ne connaissons pas le champ distance exact de cette géométrie, nous n'avons donc pas effectué d'analyse de convergence de l'erreur.



**Figure 5.12:** Champ distance du *Teapot*.

Le tableau 5.6(d) résume différentes statistiques sur le calcul du champ distance du *Utah Teapot*.



**Figure 5.13:** Surfaces iso pour différentes valeurs  $\phi$  du champ distance du *Teapot*.

## 5.6 Performance et temps de calcul

Les temps nécessaires pour calculer le champ distance des modèles géométriques préalables pour différentes tailles de la grille sont présentés dans le tableau 5.6 et sur le graphique 5.6. On observe que la relation entre le temps de calcul et le nombre de nœuds est linéaire. Le temps de calcul du champ distance d'un solide paramétrique est en  $O(h)$ . Les calculs ont été effectués sur un *MacPro*<sup>©</sup> muni de deux processeurs *Intel*<sup>©</sup> *Dual-Core Xeon*<sup>©</sup> cadencés à 2.66 GHz avec une mémoire partagée de 5 GB. Le programme, écrit en C++ et compilé à l'aide de *gcc*, n'a pas été développé pour fonctionner sur une architecture multiprocesseurs. Ainsi un seul cœur a été utilisé lors des tests.

Les temps de calcul proviennent du système d'exploitation. Nous avons mesuré le temps d'utilisation (*user time*). Le temps d'utilisation permet de déterminer précisément le temps réellement utilisé par une application, en excluant toute les interventions du système (changement de contexte, défaut de pages, etc.).

À notre connaissance, il n'existe pas de travaux similaires qui calculent le champ distance d'un solide paramétrique. Notre méthode de calcul est plus lente que les méthodes utilisant une géométrie triangulée et les méthodes des estimateurs locaux par chanfrein ou vectoriels. Mais la complexité des calculs mis en œuvre par le modèle paramétrique n'est absolument pas comparable aux modèles discrets. Pour une géométrie discrète, aucune évaluation de la surfaces ou des ses dérivées n'est nécessaire.

## 5.7 Résumé

Dans un premier temps, nous avons validé l'implémentation du *Fast Sweeping* et nous avons montré que les résultats obtenus étaient similaires à ceux publiés. Dans un deuxième temps, nous avons présenté différents modèles géométriques permettant de valider notre méthode de calcul du champ distance d'un solide paramétrique. Pour l'ensemble des tests effectués, les analyses d'erreur ont montré que la précision converge avec la taille de la grille, conformément à nos attentes. La détec-

**Tableau 5.6:** Performance et temps de calcul.

(a) Sphère (coefficient d'expansion de 10 %).

$h$	0.12	0.06	0.03	0.015
Nombre de nœuds total	8000	64,000	512,000	4,096,000
Nombre de nœuds frontières	2650	11,001	43,858	174,895
Optimisations effectuées	10,947	35,256	115,644	424,192
Temps calcul des conditions frontières (s)	54.53	224.35	871.34	4806.67
Temps calcul total (s)	54.54	224.41	871.85	4810.71

(b) Cylindre (coefficient d'expansion de 10 %).

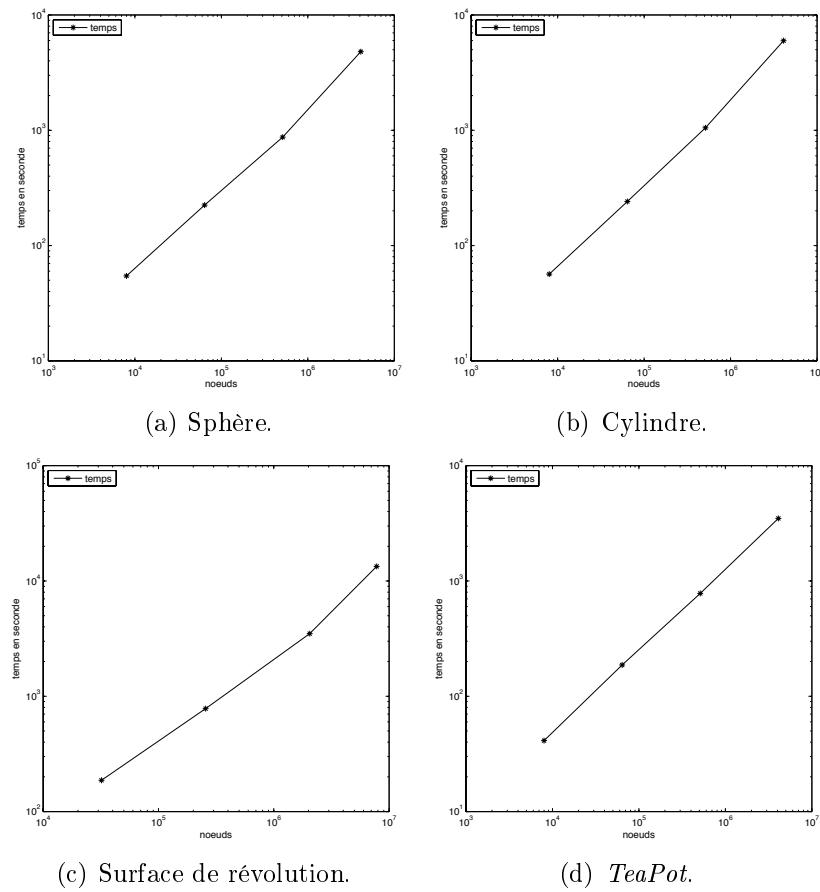
$h$	0.12	0.06	0.03	0.015
Nombre de nœuds total	8000	64,000	512,000	4,096,000
Nombre de nœuds frontières	3818	16,130	64,418	259,414
Optimisations effectuées	13248	46,142	154,634	560,936
Temps calcul des conditions frontières (s)	56.55	241.05	1050.24	5982.1
Temps calcul total (s)	56.56	241.11	1050.74	5986.19

(c) Surface de révolution (coefficient d'expansion de 20 %).

$h$	0.14	0.07	0.035	0.0224
Nombre de nœuds total	32,000	256,000	2,048,000	7,812,500
Nombre de nœuds frontières	6662	26,964	118,557	283,278
Optimisations effectuées	32,811	103,141	324,773	815,684
Temps calcul des conditions frontières (s)	186.86	780.32	3491.76	13,373.7
Temps calcul total (s)	186.89	780.57	3493.82	13,381.8

(d) Teapot (coefficient d'expansion de 20 %).

$h$	0.12	0.06	0.03	0.015
Nombre de nœuds total	8000	64,000	512,000	4,096,000
Nombre de nœuds frontières	1625	6662	26,964	118,557
Optimisations effectuées	6850	19,961	55,050	165,097
Temps calcul des conditions frontières (s)	41.24	186.86	780.32	3490.09
Temps calcul total (s)	41.24	186.89	780.57	3492.14



**Figure 5.14:** Graphique du temps nécessaire pour calculer le champ distance de différents solides paramétriques. On observe que le temps de calcul augmente linéairement avec le nombre de noeuds de la grille.

tion et l'orientation des nœuds frontières ont été validés à l'aide de modèles géométriques dont le champ distance est connu analytiquement. L'orientation du champ distance dépend de la robustesse du modèle de représentation utilisé. L'orientation géométrique à elle seule ne suffit pas à déterminer si un point est situé à l'intérieur ou à l'extérieur d'un solide.

## CONCLUSION

### Réalisations et contributions

Nous avons présenté une méthode novatrice de calcul du champ distance d'un solide paramétrique. Bien que de nombreux travaux aient déjà été réalisés sur le champ distance d'une géométrie polygonale, aucune solution n'avait encore été apportée pour obtenir celui d'une géométrie paramétrique.

Le champ distance est calculé numériquement en résolvant une équation de transport basée sur l'équation Eikonal. L'équation Eikonal est une équation aux dérivées partielles du premier ordre, dites d'Hamilton-Jacobi, et la résolution de cette équation est un problème hyperbolique non linéaire. Notre méthode repose sur le transport de conditions limites qui dépendent de la géométrie. L'information se propage le long des courbes caractéristiques à partir des conditions limites. Le caractère non linéaire du problème conduit à la formation de chocs selon la loi de conservation hyperbolique.

La méthode du *Fast-Sweeping* développée par Zhao est une méthode de balayage. Comme nous l'avons déjà expliqué, cette méthode a été choisie de préférence à une méthode d'avance de front telle que le *Fast-Marching* pour des raisons de performance.

Le *Fast-Sweeping* consiste à discréteriser l'espace autour du solide à l'aide d'une grille cartésienne puis à propager les conditions frontières en résolvant numériquement l'équation Eikonal selon un schéma de Godunov. La méthode du *Fast-Sweeping* peut être subdivisée en deux étapes. Dans un premier temps, il est nécessaire d'initialiser les conditions frontières. Dans un deuxième temps, en utilisant plusieurs balayages alternés de Gauss-Seidel, on résout localement l'équation Eikonal.

L'initialisation des conditions frontières est l'étape la plus importante et aussi la plus difficile à mettre en œuvre lors du calcul du champ distance d'un solide. Ces conditions limites sont constituées des nœuds limitrophes à la géométrie, appelés *noeuds frontières*.

Il a été nécessaire de développer un algorithme permettant de détecter et d'initialiser précisément les noeuds frontières. Ces noeuds frontières sont initialisés avec la valeur de la plus courte distance euclidienne permettant de rejoindre le solide. Cette distance minimale est obtenue en mesurant la longueur entre le noeud et sa projection orthogonale sur la surface du solide.

Une méthode précise de projection sur une surface de Bézier a été développée. Contrairement aux méthodes communément admises, qui calculent la projection à l'aide d'un critère d'orthogonalité, nous avons développé une nouvelle approche basée sur la minimisation d'une distance euclidienne.

Finalement, dans le cas d'un solide fermé, l'orientation du champ distance est déterminée en propagant l'orientation des noeuds frontières. La localisation des noeuds intérieurs et des noeuds extérieurs est déterminée grâce de l'orientation géométrique de la surface.

Notre méthode de calcul du champ distance a été validée en effectuant une analyse de convergence de l'erreur à partir de solides dont le champ distance exact est connu. La robustesse de l'orientation du champ distance a, quant à elle, été validée à l'aide de plusieurs géométries où la localisation d'un point de l'espace par rapport à la surface est connue sans ambiguïté.

Une des visées de cette recherche était l'élaboration d'une méthode de calcul du champ distance qui conserverait la continuité du modèle paramétrique original. Les méthodes développées jusqu'à présent ne s'appliquent qu'aux surfaces polygonales. Comme nous l'avions précisé auparavant, le caractère non linéaire de l'équation Eikonal conduit à la formation de zones d'expansion au niveau des arêtes et des sommets d'une triangulation. Nos résultats nous permettent d'affirmer que la méthode que nous avons proposée permet de palier ce problème. Le champ distance obtenu à partir d'un modèle paramétrique est beaucoup plus régulier que celui obtenu pour une géométrie équivalente mais triangulée.

Un aspect important pour répandre l'usage du champ distance comme un modèle de représentation géométrique viable était de garantir une bonne performance. Nous avions comme objectif de proposer une méthode dont le temps de calcul serait proportionnel à la difficulté du problème. Les mesures effectuées ont montré que les

temps de calcul des différents algorithmes développés sont linéaires par rapport à la discréétisation utilisée. L'étape la plus longue est la construction des conditions limites. La détection et l'initialisation des nœuds frontières font massivement appel à un algorithme d'optimisation. La librairie d'optimisation que nous avons utilisée n'est pas adaptée à un usage aussi intensif, ce qui ralentit considérablement notre méthode de calcul. Cependant les résultats obtenus sont considérablement supérieurs à ceux publiés dans la littérature. Des champs distances supérieurs à sept millions de nœuds ont été obtenus dans un laps de temps raisonnable comparativement aux problèmes de dimension très inférieure des méthodes concurrentes.

## Limites et contraintes

Certaines limites sont apparues pendant nos tests. Premièrement, nous nous sommes aperçu que la projection, comme toute méthode d'optimisation, ne donne pas un résultat exact. La précision est liée à la nature de la fonction, dans notre cas à la paramétrisation de la surface. Celle-ci, lorsqu'elle est dégénérée rend la projection plus délicate. L'erreur maximale  $\|e\|_\infty$  semble être constante quelle que soit la surface. Cette constatation à une implication directe sur la discréétisation minimale  $h_{\min}$  que l'on peut choisir. Il existe donc une relation telle que  $h_{\min} > \beta\|e\|_\infty$  à déterminer. Cette contrainte impose une limite à la précision maximale qui peut être attendue de notre méthode de calcul du champ distance.

Deuxièmement, pour certaines géométries, l'orientation des noeuds peut être erronée. Notre méthode de localisation, basée sur le calcul de normales, n'est pas toujours robuste. Bien qu'il soit évident pour l'œil humain de situer rapidement un point à l'intérieur ou à l'extérieur d'un solide, c'est un problème difficile en géométrie. Nous avions rencontré des problèmes similaires lors de nos travaux préalables sur l'orientation du champ distance des modèles polygonaux. À la lumière de nos résultats, l'utilisation d'une géométrie paramétrique ne semble pas permettre, pour l'instant, de garantir une orientation exacte quelle que soit la géométrie.

## Amélioration

Notre reconstruction implicite d'un solide paramétrique nous a apporté satisfaction, mais elle est perfectible.

D'après nos expérimentations, il semble que le calcul des normales soit une étape cruciale, aussi bien lors de l'optimisation que lors de l'orientation des conditions limites. Une méthode d'évaluation robuste des normales permettrait d'améliorer nos résultats. Les travaux de Kiciak (2001) concernant la construction de surfaces tangentes de surfaces normales seraient une approche intéressante. En plus de permettre une reconstruction des normales dégénérées, l'évaluation des dérivées est très performante en dehors du coût marginal de construction des surfaces tangentes. Cette méthode nous permettrait de faire d'une pierre deux coups, en améliorant l'évaluation de tangentes lors de l'optimisation d'une part et en rendant plus robuste l'orientation des nœuds frontières d'autre part.

Cependant, dans certains cas particuliers où il y a une ambiguïté sur la surface la plus proche d'un nœud, la normale ne permet pas de déterminer l'orientation du nœud. Il serait judicieux de compléter notre méthode de localisation avec une méthode de lancer de rayon. Cette méthode présente l'avantage indéniable de n'avoir aucune restriction sur le rayon. Ainsi, s'il y a un doute sur le résultat, on peut simplement effectuer un autre lancer de rayon dans une autre direction. Néanmoins, le calcul des intersections entre un rayon et un solide est, lui aussi, complexe, particulièrement lors que le point d'interrogation est proche de la surface, ce qui est notre cas.

Ce travail soulève aussi de nombreuses questions qui méritent des études particulières. Afin de proposer une méthode automatique d'implicitisation d'un modèle paramétrique, il serait souhaitable de déterminer quelle est la discrétisation  $h$  maximale permettant de représenter adéquatement un solide. L'utilisateur doit, pour l'instant, décider empiriquement de la taille de la grille en fonction de la précision requise et des particularités de la géométrie. Une méthode systématique d'analyse des propriétés de la surface permettant d'estimer  $h$  reste à déterminer.

## Travaux futurs

Jusqu'à présent, le temps nécessaire pour calculer un champ distance était une contrainte qui, a semble-t-il, freiné son usage et le développement de nouvelles applications. Avec l'avènement des méthodes de balayages rapides, couplées avec des estimateurs locaux plus précis, le champ distance devient un outil attrayant.

La première étape qui devrait suivre nos travaux, est le développement d'une méthode d'interpolation et de reconstruction du gradient. L'interpolation du champ distance n'est pas triviale en raison des chocs présents dans la solution. Notamment, la reconstruction du gradient est complexe. Le gradient n'est pas défini à l'intersection des caractéristiques et malheureusement, la détection de ces singularités est très importante dans la plupart des applications utilisant un champ distance. Le gradient ne peut pas être calculé par de simples différences finies. Afin de calculer correctement le gradient, il faut utiliser un interpolant qui respecte le sens de l'écoulement des caractéristiques. Les méthodes ENO (*Essentially Non Oscillatory*) permettent d'interpoler le gradient au niveau des chocs et à proximité de la surface.

Une fois qu'une méthode d'interpolation aura été développée, il sera possible de concevoir une méthode permettant de traverser le champ distance, en suivant l'écoulement des caractéristiques, et de détecter les chocs. Il s'agit de se déplacer dans la direction du gradient, en intégrant à l'aide d'une méthode numérique comme celle de Runge-Kutta. Afin de contrôler le déplacement, les méthodes d'intégration numériques utilisent des critères d'arrêts : c'est là que la détection des chocs est importante.

La décomposition de domaine automatique est encore au stade de la recherche. Un découpage selon l'axe médian et selon les régions de Voronoï semble être l'un des fondements des techniques de décomposition automatique. Le champ distance permet de localiser les axes médians. Il permet aussi de détecter les frontières des polytopes convexes qui définissent les régions de Voronoï. Rappelons qu'une triangulation de Delaunay est le dual d'une partition de Voronoï. Par conséquent, il existe un lien étroit entre la détection des singularités d'un champ distance et la décomposition d'un domaine.

Une des particularités du champ distance est de contenir implicitement toutes les isosurfaces possibles.

Une autre façon de décomposer un domaine est de construire des surfaces paramétriques tubulaires. Ces surfaces permettent de construire des domaines fermés par quatre frontières paramétriques. Un maillage peut ensuite être généré à l'aide de méthodes elliptiques. La construction de ces nouvelles frontières paramétriques demande de développer des méthodes de paramétrisation de surfaces implicites.

Le calcul du champ distance met en œuvre d'importantes ressources en termes de temps de calcul et d'espace mémoire. Si un champ distance plus précis est requis, raffiner de façon globale la grille cartésienne peut être impossible, car l'augmentation du nombre de nœuds augmente rapidement et l'espace mémoire disponible peut devenir insuffisant. Il existe une méthode de calcul du champ distance d'ordre supérieur. Elle permet d'obtenir une solution plus précise sans augmenter l'espace mémoire. Il serait intéressant d'utiliser cette méthode, à condition que la précision des conditions frontières soit suffisante.

Il serait nécessaire de construire des surfaces paramétriques de référence propres au domaine de la géométrie numérique. Nous n'avons pas été en mesure de trouver dans la littérature des surfaces paramétriques pour lesquelles une analyse complète aurait été fournie. Pendant le développement de la méthode de projection, nous aurions eu besoin de surfaces de référence pour valider plus formellement nos résultats et vérifier plus facilement l'implémentation des algorithmes de calculs. Ces surfaces de référence seraient un outil similaire aux 306 tests de Hock et Schittkowski pour les méthodes d'optimisation. Pour chacune de ces surfaces, nous devrions disposer de plusieurs formes de représentations (B-splinaire, décomposition en Bézier), de paramétrisations différentes, l'évaluation de la surface et de ses dérivées en certains points particuliers, la localisation des singularités, etc. Plusieurs surfaces seraient certainement nécessaires pour couvrir l'ensemble des cas pouvant poser problèmes (tangentes dégénérées, nœuds du polygone de contrôle confondus, etc.). Avec de telles surfaces, la comparaison entre les différents travaux traitant de la projection serait facilitée, notamment au niveau de la précision et de la performance.

Le champ distance est un modèle de représentation implicite prometteur qui mérite

d'être plus amplement utilisé. Bien que les calculs permettant d'obtenir le champ distance d'un solide paramétrique puissent paraître lourds, la représentation implicite d'un solide, en particulier le champ distance, permet une approche différente de la modélisation géométrique. Sans vouloir remplacer les modèles paramétriques, nous croyons que le champ distance est une représentation complémentaire qui offre de nombreux avantages, notamment dès que des notions de distances et de surfaces iso entrent en jeu.

## BIBLIOGRAPHIE

- AUMANN, G., REIF, U. et SPITZMULLER, K. (1995). A counterexample to a corollary of Kim et al. *Computer aided geometric design*, 12, 853–855.
- BARTH, T. J. et SETHIAN, J. A. (1998). Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains. *J. Comput. Phys.*, 145, 1–40.
- BOUIX, S. et SIDDIQI, K. (2000). Divergence-based medial surfaces. *ECCV '00 : Proceedings of the 6th European Conference on Computer Vision-Part I*. Springer-Verlag, London, UK, 603–618.
- BUTT, M. et MARAGOS, P. (1998). Optimum design of chamfer distance transforms. *IEEE Transactions on image processing*, 7, 1477.
- CHEN, X., SU, H., YONG, J., PAUL, J. et SUN, J. (2007). A counterexample on point inversion and projection for NURBS curve. *Computer Aided Geometric Design*, 24, 302–302.
- CHENG, F. (1992). Estimating subdivision depths for rational curves and surfaces. *ACM Trans. Graph.*, 11, 140–151.
- CRANDALL, M. et LIONS, P. (1983). Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 1–42.
- EDELSBRUNNER, H. et MÜCKE, E. P. (1990). Simulation of simplicity : a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9, 66–104.
- ELIAS, R., MARTINS, M. et COUTINHO, A. (2007). Simple finite element-based computation of distance functions in unstructured grids. *International Journal for Numerical Methods in Engineering*, 72, 1095–1110.
- FARIN, G. (1993). *Curves and Surfaces for CAGD : A Practical Guide*. Morgan Kaufmann.
- FREY, P. J. et GEORGE, P.-L. (1999). *Maillages. Applications aux éléments finis*. Hermès, Paris.

- GREMAUD, P. A. et KUSTER, C. M. (2006). Computational study of fast methods for the eikonal equation. *SIAM J. Sci. Comput.*, 27, 1803–1816.
- GRIGOR'EV, M. I. (2008). Construction of a sphere by using projective Bézier patches. *Vestnik St. Petersburg University : Mathematics*, 41, 273–277.
- HU, S.-M. et WALLNER, J. (2005). A second order algorithm for orthogonal projection onto curves and surfaces. *Computer Aided Geometric Design*. 251–260.
- JONES, M. W. (1995). 3d distance from a point to a triangle. Rapport technique CSR-5-95, Department of Computer Science, University of Wales Swansea.
- JONES, M. W., BAERENTZEN, J. A. et SRAMEK, M. (2006). 3D Distance Fields : A Survey of Techniques and Applications. *IEEE Transactions on Visualization and Computer Graphics*, 12, 581–599.
- KAO, C., OSHER, S. et QIAN, J. (2004). Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations. *Journal of Computational Physics*, 196, 367 – 391.
- KICIAK, P. (2001). Computing normal vector Bézier patches. *Computer Aided Geometric Design*, 18, 699–710.
- KIM, D., PAPALAMBROS, P. et WOO, T. (1995). Tangent, normal, and visibility cones on bezier surfaces. *Computer Aided Geometric Design*, 12, 305–320.
- KRANTZ, S. et PARKS, H. (1981). Distance to  $c^k$  hypersurfaces. *J. Differential Equations*, 40, 116–120.
- LORENSEN, W. E. et CLINE, H. E. (1987). Marching cubes : A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21, 163–169.
- LUENBERGER, D. G. (1997). *Optimization by Vector Space Methods*. Wiley, New York.
- MA, Y., HEWITT, W. et TURNER, M. (2006). A High-Performance Method for Calculating the Minimum Distance Between Two 2D and 3D NURBS Curves. *JOURNAL OF GRAPHICS TOOLS*, 11, 37.

- MA, Y. et HEWITT, W. T. (2003). Point inversion and projection for NURBS curve : Control polygon approach. *TPCG '03 : Proceedings of the Theory and Practice of Computer Graphics 2003*. IEEE Computer Society, Washington, DC, USA, 113.
- MARCHANDISE, E., REMACLE, J.-F. et CHEVAUGEON, N. (2006). A quadrature-free discontinuous galerkin method for the level set equation. *J. Comput. Phys.*, 212, 338–357.
- MEZA, J. C., OLIVA, R. A., HOUGH, P. D. et WILLIAMS, P. J. (2007). Opt++ : An object-oriented toolkit for nonlinear optimization. *ACM Trans. Math. Softw.*, 33, 12.
- MORA, B., JESSEL, J. et CAUBET, R. (2000). Accelerating volume rendering with quantized voxels. *Proceedings of the 2000 IEEE symposium on Volume visualization*.
- MOURAD, H., DOLBOW, J. et GARIKIPATI, K. (2005). An assumed-gradient finite element method for the level set equation. *Int J Numer Methods Eng*, 64, 1009–1032.
- OSHER, S. et SETHIAN, J. A. (1988). Fronts propagating with curvature-dependent speed : Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79, 12–49.
- PATRIKALAKIS, N. M. (2002). *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- PAYNE, B. A. et TOGA, A. W. (1992). Distance field manipulation of surface models. *IEEE Comput. Graph. Appl.*, 12, 65–71.
- PERSSON, P. (2004). *Mesh Generation for Implicit Geometries*. Thèse de doctorat, Massachusetts Institute of Technology.
- PFISTER, H. (1999). Architectures for real-time volume rendering. *Future Generation Computer Systems*, 15, 1–9.
- PIEGL, L. et TILLER, W. (1995). *The NURBS Book*. Springer.

- SATHERLEY, R. et JONES, M. (2001a). Vector-city vector distance transform. *Computer Vision and Image Understanding*, 82, 238–254.
- SATHERLEY, R. et JONES, M. W. (2001b). Hybrid distance field computation. *Volume Graphics 2001*. Springer, 195–209.
- SCHNEIDER, P. J. et EBERLY, D. (2002). *Geometric Tools for Computer Graphics*. Elsevier Science Inc., New York, NY, USA.
- SEGURA, R. et FEITO, F. (1998). An algorithm for determining intersection segment-polygon in 3D. *Computer and Graphics*, 22, 587–592.
- SELIMOVIC, I. (2006). Improved algorithms for the projection of points on NURBS curves and surfaces. *Computer Aided Geometric Design*, 23, 439–445.
- SEOUL, S. (1995). Detection of degenerate normal vectors on parametric surfaces : Tangent cone approach. *Computer Aided Geometric Design*, 12, 321–327.
- SETHIAN, J. A. (1999). *Level set methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press.
- SHEWCHUK, J. R. (1997). Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete and Computational Geometry*. vol. 18, 305–363.
- SUSSMAN, M. et FATEMI, E. (1999). An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20, 1165–1191.
- TSAI, Y.-H. R. (2002). Rapid and accurate computation of the distance function using grids. *J. Comput. Phys.*, 178, 175–195.
- TSITSIKLIS, J. (1995). Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40, 1528–1538.
- VELHO, L. et GOMES, J. (1996). Approximate conversion of parametric to implicit surfaces. *Computer Graphics Forum*, 15, 327–337.

- WANG, Y. L., GUIBAULT, F., CAMARERO, R. et TCHON, K.-F. (2005). Eikonal equation-based offset of curves and surfaces. SIAM Conference on Computational Science Engineering.
- YAMAGUCHI, Y. (1997). Bezier normal vector surface and its applications. *Shape Modeling and Applications, 1997. Proceedings., 1997 International Conference on.* 26–35.
- ZHANG, Y.-T., ZHAO, H.-K. et QIAN, J. (2006). High order fast sweeping methods for static hamilton—jacobi equations. *J. Sci. Comput., 29,* 25–56.
- ZHAO, H. (2005). Fast Sweeping Method for Eikonal Equations. *Mathematics of Computation, 74,* 603–627.

## ANNEXE I

### PROJECTION SUR UNE FACETTE TRIANGULAIRE

#### I.1 Distance minimale d'un point à un triangle

Plusieurs méthodes ont été développées pour calculer la plus courte distance entre un point de l'espace et un triangle. Jones (1995) compare deux méthodes. La première méthode consiste utiliser une projection géométrique, alors que la seconde transforme le problème tridimensionnel en un problème bidimensionnel.

Les résultats publiés dans Jones (1995) montrent que la méthode en 2D est trois à quatre fois plus rapide que la méthode par projection en 3D. Cependant il est possible de calculer la distance minimale d'un point à un triangle de façon analytique et ainsi éviter les difficultés géométriques liées à la localisation d'un point par rapport à un triangle.

#### I.2 Distance calculée analytiquement

Afin d'éviter les problèmes de robustesse des approches géométriques, on préfère utiliser la méthode analytique publiée dans Schneider et Eberly (2002).

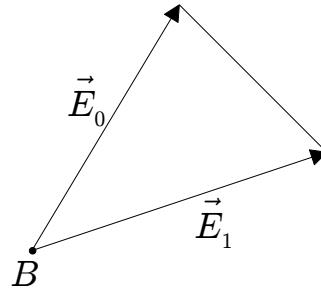
Soit un point  $\vec{P}$  et un triangle  $\vec{T}(s, t) = \vec{B} + s\vec{E}_0 + t\vec{E}_1$  tel que :

$$T(s, t) \in D = \{(s, t) : s \in [0, 1], t \in [0, 1], s + t \leq 1\}.$$

La distance minimale est calculée en déterminant les valeurs  $(\bar{s}, \bar{t}) \in D$  correspondant au point du triangle le plus proche de  $\vec{P}$ .

Soit  $Q(s, t)$  la fonction donnant la distance au carré entre le point  $\vec{P}$  et  $\vec{T}(s, t)$  définie comme :

$$Q(s, t) = |\vec{T}(s, t) - \vec{P}|^2 \text{ pour } (s, t) \in D.$$



**Figure I.1:** Illustration du triangle  $T(s, t)$ .

La fonction  $Q(s, t)$  est quadratique en  $s$  et  $t$  :

$$Q(s, t) = as^2 + 2bst + ct^2 + 2ds + 2et + f.$$

où  $a = \vec{E}_0 \cdot \vec{E}_0$ ,  $b = \vec{E}_0 \cdot \vec{E}_1$ ,  $c = \vec{E}_1 \cdot \vec{E}_1$ ,  $d = \vec{E}_0 \cdot (\vec{B} - \vec{P})$ ,  $e = \vec{E}_1 \cdot (\vec{B} - \vec{P})$ ,  $f = (\vec{B} - \vec{P}) \cdot (\vec{B} - \vec{P})$ .

La forme d'une quadratique dépend du signe de  $ac - b^2$ .

Pour la fonction  $Q$ ,

$$ac - b^2 = (\vec{E}_0 \cdot \vec{E}_0)(\vec{E}_1 \cdot \vec{E}_1) - (\vec{E}_0 \cdot \vec{E}_1)^2 = |\vec{E}_0 \times \vec{E}_1|^2 > 0.$$

Le signe  $ac - b^2$  est toujours positif, car on suppose que les deux côtés  $\vec{E}_0$  et  $\vec{E}_1$  du triangle sont linéairement indépendants, donc que leur produit vectoriel n'est pas nul.

Déterminer le point du triangle le plus proche de  $P$  revient à minimiser la fonction  $Q(s, t)$  sur le domaine  $D$ . Comme la fonction  $Q(s, t)$  est continue et différentiable, le minimum est soit un point à l'intérieur de  $D$ , tel quel le gradient  $\nabla \vec{Q} = (0, 0)$ , ou un point sur la frontière de  $D$ . Le gradient de la fonction  $Q(s, t)$  est :

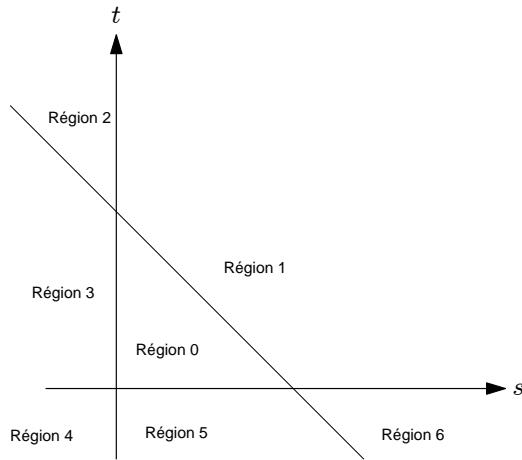
$$\nabla \vec{Q} = 2(as + bt + d, bs + ct + e).$$

Le gradiant de la fonction  $Q(s, t)$  est nul lorsque

$$\bar{s} = \frac{be - cd}{ac - b^2}.$$

$$\bar{t} = \frac{bd - ae}{ac - b^2}.$$

Si  $(\bar{s}, \bar{t}) \in D$ , alors le point recherché est dans le triangle. Si ce n'est pas le cas, alors le point recherché est sur la frontière de  $T$ . Afin de localiser correctement le minimum, on divise le plan  $st$  en sept régions distinctes comme le montre la figure I.2.

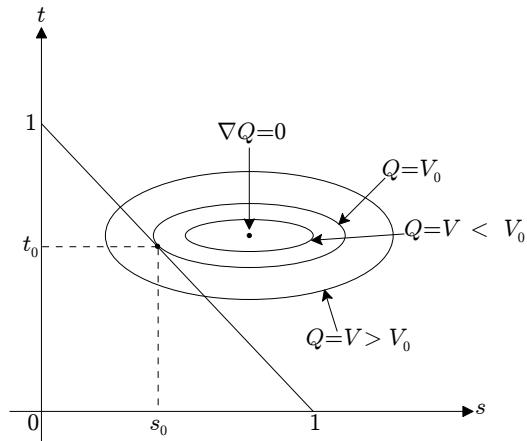


**Figure I.2:** Division du plan  $st$  afin de localiser le minimum lorsqu'il est situé sur une arête ou un sommet du triangle.

La région centrale nommée *Région 0* sur la figure I.2, est le domaine de  $Q(s, t)$ . Si  $(\bar{s}, \bar{t})$  est situé dans la *Région 0*, le minimum de la fonction  $Q$  est dans  $D$ , alors le point le plus proche de  $P$  est situé à l'intérieur du triangle. Cependant, il est possible que le minimum global de  $Q$  ne soit pas dans  $D$ . Il est alors nécessaire de déterminer la plus petite valeur  $(\bar{s}, \bar{t})$  de  $Q$  tel que  $(\bar{s}, \bar{t})$  soit dans  $D$ . On doit donc traiter plusieurs cas pour chaque région du domaine.

Supposons maintenant que le point  $(\bar{s}, \bar{t})$  soit situé dans la *Région 1*. Les courbes de niveaux de la fonction paraboloïde  $Q$  forment dans le plan  $st$  des ellipses. Au point où le gradiant de  $Q$  est nul :  $\nabla Q = (0, 0)$ , les courbes de niveaux dégénèrent en un point  $(\bar{s}, \bar{t})$ . Ce point est le minimum global de  $Q$  et a pour valeur  $V_{min}$ .

Lorsque la valeur seuil  $V$  des courbes de niveaux augmente par rapport à  $V_{min}$ , les ellipses s'éloignent du minimum  $(\bar{s}, \bar{t})$ . Il existe une valeur  $V_0$  pour laquelle la courbe de niveau correspondante est tangente au segment  $s + t = 1$ , formant une arête du triangle, au point  $s_0 \in [0, 1]$ ,  $t_0 = 1 - s_0$ . Il n'y a aucune intersection entre les courbes de niveaux et le domaine  $D$  pour une valeur seuil  $V < V_0$ . Pour toutes les courbes de niveaux  $V > V_0$ , une partie du domaine de  $D$  est situé à l'intérieur de l'ellipse.



**Figure I.3:** Courbes de niveaux de la fonction  $Q$  pour la Région 1. Le point  $(s_0, t_0) \in D$  est le minimum recherché de la fonction  $Q$ .

Tous les points formant une intersection entre une ellipse et un côté du triangle doivent avoir une valeur seuil  $V > V_0$ . On en déduit que  $Q(s, 1 - s) > Q(s_0, t_0)$  pour  $s \in [0, 1]$  et  $s \neq s_0$ . Le point  $(s_0, t_0)$  est donc la distance au carré minimale recherchée entre le point  $P$  et le triangle. La figure I.3 illustre les quatre cas possibles de valeur seuil pour les courbes de niveau de  $Q$ .

Une autre façon de visualiser le problème de minimisation de  $Q$  sur le domaine  $D$  est de considérer l'intersection entre la paraboloïde  $Q$  et le plan d'équation  $s + t = 1$ . L'intersection est formée par une parabole d'équation  $F(s) = Q(s, 1 - s)$  pour  $s \in [0, 1]$ . Le problème de minimisation de  $Q(s, t)$  peut ainsi être ramené à un problème unidimensionnel, à savoir la minimisation de  $F(s)$  sur l'intervalle  $[0, 1]$ . Le minimum de  $F(s)$  a lieu soit à l'intérieur du domaine  $[0, 1]$ , auquel cas la dérivée  $F'(s) = 0$  en ce point, soit sur une des bornes du domaine,  $s = 0$  ou  $s = 1$ .

Afin de déterminer la localisation du minimum recherché de la fonction  $F(s)$ , on

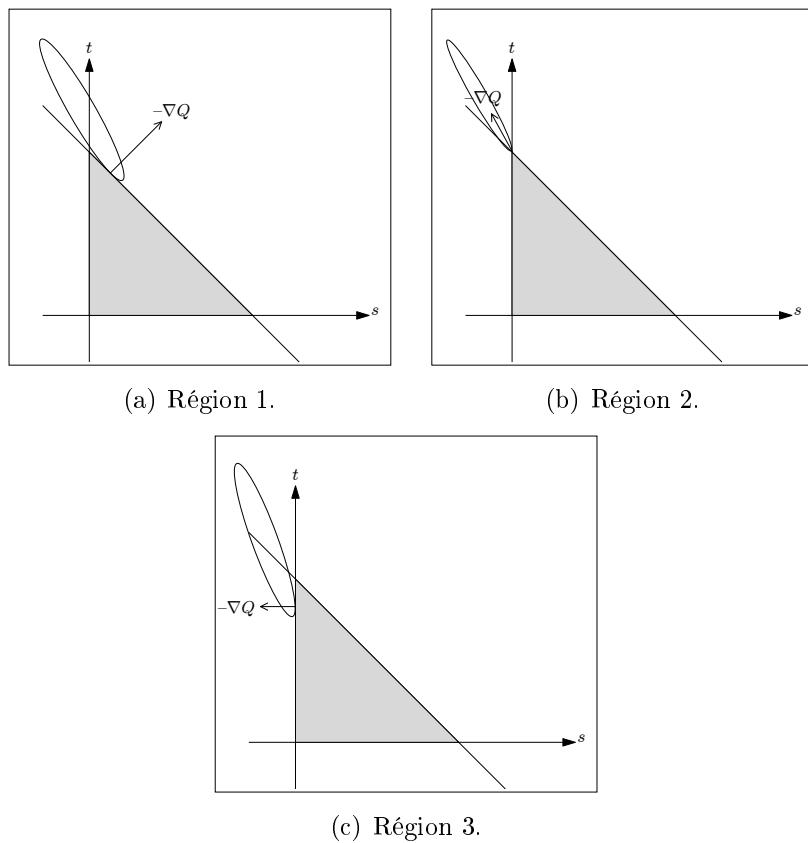
effectue une partition des valeurs réelles possible. L'axe des réels est divisé en trois intervalles :  $s < 0$ ,  $s \in [0, 1]$  et  $s > 1$ . Soit la valeur  $s^*$  tel que  $F'(s^*) = 0$  :

- Si  $s^* < 0$ , alors la fonction  $F(s)$  est croissante sur l'intervalle  $[0, 1]$  et le minimum recherché sur l'intervalle à donc lieu au point  $s = 0$ . Le minimum recherché de  $Q$  sur  $D$  est donc le point  $(s, t) = (0, 1)$  ;
- Si  $s^* > 0$ , alors la fonction  $F(s)$  est décroissante sur l'intervalle  $[0, 1]$  et le minimum recherché sur l'intervalle à donc lieu au point  $s = 1$ . Le minimum recherché de  $Q$  sur  $D$  est donc le point  $(s, t) = (1, 0)$  ;
- Si  $s^* \in [0, 1]$  alors la fonction  $Q$  atteint son minimum au point  $(s, t) = (s^*, 1 - s^*)$ . Si le minimum global  $(\bar{s}, \bar{t})$  de la fonction  $Q$  survient dans la région 3 ou 5, on traite le cas de façon similaire à la région 0 :
- Si  $(\bar{s}, \bar{t})$  est dans la région 3, le minimum recherché est un point  $(0, t_0)$  pour  $t_0 \in [0, 1]$  ;
- Si  $(\bar{s}, \bar{t})$  est dans la région 5, le minimum recherché est un point  $(s_0, 0)$  pour  $s_0 \in [0, 1]$ .

On détermine si le premier point de contact est une borne ou un point intérieur du domaine de la même manière que précédemment.

Si  $(\bar{s}, \bar{t})$  est situé dans la région 2, l'intersection entre la courbe de niveau et le domaine  $D$  peut avoir lieu au sommet  $(0, 1)$ , sur l'arête  $s = 0$  ou bien sur l'arête  $s + t = 1$ . La figure I.2 illustre les trois cas possibles.

Le gradient  $-\nabla Q$  indique la direction intérieure de la courbe de niveau ; ainsi, le gradient ne pointe jamais vers l'intérieur du triangle. On utilise cette propriété afin de déterminer dans laquelle des trois possibilités précédentes nous nous situons. Les deux vecteurs  $(0, 1)$  et  $(1, -1)$  donnent la normale pour les côtés  $s = 0$  et  $s + t = 1$  respectivement. Le gradient  $\nabla Q = (Q_s, Q_t)$ , où  $Q_s$  et  $Q_t$  sont les dérivées partielles de  $Q$ , permet de calculer les deux produits scalaires  $(0, 1) \cdot \nabla Q(0, 1)$  et  $(-1, 1) \cdot \nabla Q(0, 1)$ . Ces deux produits scalaires ne peuvent pas avoir simultanément le même signe. Ainsi on peut localiser sur quel côté à lieu l'intersection à l'aide du signe des gradients  $(0, 1) \cdot \nabla Q(0, 1)$  et  $(-1, 1) \cdot \nabla Q(-1, 1)$ .



**Figure I.4:** Courbe de niveau possible pour la région 2.

On utilise le même procédé pour la région 6. Pour la région 4, le signe des gradients à utiliser est  $(1, 0) \cdot \nabla Q(0, 0)$  et  $(0, 1) \cdot \nabla Q(0, 0)$  .

## ANNEXE II

### SURFACES *NURBS*

#### II.1 Propriétés des surfaces *NURBS*

**Propriété 1 (Interpolation)** *Les coins du polygone de contrôle sont interpolés par la surface NURBS. Ainsi  $\vec{S}(0,0) = \vec{P}_{0,0}$ ,  $\vec{S}(1,0) = \vec{P}_{n,0}$ ,  $\vec{S}(0,1) = \vec{P}_{0,m}$  et  $\vec{S}(1,1) = \vec{P}_{n,m}$ .*

**Propriété 2 (Invariance affine)** *On applique une transformation affine à la surface NURBS en appliquant la transformation au polygone de contrôle de la surface. Les surfaces NURBS sont définies indépendamment d'un repère fixe.*

L'invariance par transformation affine (translation, rotation, homothétie) implique que la somme des valeurs des fonctions de base pour un paramètre  $u$  donné soit égale à l'unité :

$$\sum_{i=1}^n N_{i,p}^i = 1, \quad u \in [u_0; u_m].$$

**Propriété 3 (Convexité)** *Dans le cas où  $w_{i,j} \geq 0$  pour tout les  $i, j$  si  $(u, v) \in [u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]$ , alors  $\vec{S}(u, v)$  est strictement contenu dans le polygone convexe formé par les points de contrôle  $\vec{P}_{i,j}$ ,  $i_0 - p \leq i \leq i_0$  et  $j_0 - q \leq j \leq j_0$ .*

**Propriété 4 (Support local)** *Si le point  $P_{i,j}$  est déplacé ou si la pondération  $w_{i,j}$  est changée, seule la partie de la surface définie par le rectangle  $[u_i, u_{i+p+1}] \times [v_i, v_{i+q+1}]$  est modifiée.*

**Propriété 5 (Ordre des dérivées)** *La surface NURBS  $\vec{S}(u, v)$  est dérivable  $p - k$  fois selon la direction  $u$  et  $q - k$  fois dérivable selon la direction  $v$  pour une multiplicité  $k$  du vecteur nodal.*

**Propriété 6 (Diminution de variation)** *Les surfaces NURBS n'ont pas de propriété de diminution de variation.*

## II.2 Dérivées de surfaces *NURBS*

Le calcul des dérivées des surfaces *NURBS* est très important. En effet, la dérivée d'une surface intervient dans la plupart des méthodes d'analyse et de caractérisation de la surface.

Les dérivées des surfaces *NURBS* sont calculées à l'aide des dérivées des fonctions de base B-splinaires. La dérivée d'une surface B-splinaire (non rationnelle) est donnée par la formule suivante :

$$\frac{\partial^{k+l}}{\partial^k u \partial^l v} \vec{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}^{(k)} N_{j,q}^{(l)} \vec{P}_{i,j} \quad 0 \leq k + l \leq d.$$

On exprime la surface *NURBS* sous la forme rationnelle suivante :

$$\vec{S}(u, v) = \frac{w(u, v) \vec{S}(u, v)}{w(u, v)} = \frac{\vec{A}(u, v)}{w(u, v)}.$$

Ainsi, la dérivée pour le paramètre  $\alpha$  (c'est-à-dire  $u$  ou  $v$ ) s'écrit :

$$\vec{S}_\alpha(u, v) = \frac{\vec{A}_\alpha(u, v) - w_\alpha(u, v) \vec{S}(u, v)}{w(u, v)},$$

avec

$$\begin{aligned}
\vec{A}^{(k,l)} &= [(w\vec{S})^k]^l \\
&= \left( \sum_{i=0}^k \binom{k}{i} w^{(i,0)} \vec{S}^{(k-i,0)} \right)^l \\
&= \sum_{i=0}^k \binom{k}{i} \sum_{j=0}^l \binom{l}{j} w^{(i,j)} \vec{S}^{(k-i,l-j)} \\
&= w^{(0,0)} \vec{S}^{(k,l)} + \sum_{i=1}^k \binom{k}{i} w^{(i,0)} \vec{S}^{(k-i,l)} + \sum_{j=1}^l \binom{l}{j} w^{(0,j)} \vec{S}^{(k,l-j)} \\
&\quad + \sum_{i=1}^k \binom{k}{i} \sum_{j=1}^l \binom{l}{j} w^{(i,j)} \vec{S}^{(k-i,l-j)}.
\end{aligned}$$

Ainsi, la formulation générale des dérivées d'une surface *NURBS* est :

$$\vec{S}^{(k,l)} = \frac{1}{w} \left( \vec{A}^{(k,l)} - \sum_{i=1}^k \binom{k}{i} w^{(i,0)} \vec{S}^{(k-i,l)} - \sum_{j=1}^l \binom{l}{j} w^{(0,j)} \vec{S}^{(k,l-j)} - \sum_{i=1}^k \binom{k}{i} \sum_{j=1}^l \binom{l}{j} w^{(i,j)} \vec{S}^{(k-i,l-j)} \right). \quad (\text{II.1})$$

Les dérivées partielles sont donc données par :

$$\begin{aligned}
\vec{S}_{uv} &= \frac{\vec{A}_{uv} - w_{uv} \vec{S} - w_u \vec{S}_v - w_v \vec{S}_u}{w}, \\
\vec{S}_{uu} &= \frac{\vec{A}_{uu} - 2w_u \vec{S}_u - w_{uu} \vec{S}}{w}, \\
\vec{S}_{vv} &= \frac{\vec{A}_{vv} - 2w_v \vec{S}_v - w_{vv} \vec{S}}{w}.
\end{aligned}$$

### II.3 Évaluation numérique des surfaces *NURBS*

L'évaluation d'un point sur une surface *NURBS* se fait à l'aide de l'algorithme de De Casteljau. Cet algorithme récursif permet d'évaluer les polynômes exprimés dans une base de Bernstein.

Bien qu'il ait été développé à l'origine pour l'évaluation des courbes et des surfaces de Bézier, l'algorithme de De Casteljau est généralisable aux surfaces *NURBS* en utilisant les coordonnées homogènes.

Cet algorithme repose sur le fait qu'une restriction d'une surface *NURBS* est aussi une surface *NURBS*. En effet les polynômes de Bernstein possèdent la propriété de récurrence suivante :

$$B_i^n(u) = (1 - u)B_i^{n-1}(u) + uB_{i-1}^{n-1}(u).$$

Même s'il est possible d'évaluer une surface *NURBS* en calculant la forme analytique des polynômes de Bernstein, il est admis que l'algorithme de De Casteljau est numériquement plus stable.

Géométriquement, l'algorithme de De Casteljau peut être interprété comme un processus de subdivision qui raffine le polygone de contrôle jusqu'à l'obtention d'un point situé sur la courbe ou la surface. Le processus de subdivision, pour évaluer une courbe au paramètre  $t$ , consiste à déterminer, pour chaque branche du polygone de contrôle, le point situé au ratio  $\frac{t}{1-t}$ . En reliant tous les points obtenus, on obtient un nouveau polygone de contrôle ayant une branche en moins. On répète ce processus jusqu'à l'obtention d'un seul point.

Les méthodes d'évaluation sont très importantes dans ce projet en raison de l'utilisation intensive d'algorithmes d'optimisation évaluant un très grand nombre de fois la surface.

## II.4 Insertion de nœuds dans une surface *NURBS*

L'insertion de nœuds fait partie des opérations les plus importantes permettant de manipuler les surfaces B-splinaires. L'insertion de nœuds permet :

1. D'évaluer un point et ses dérivées, aussi bien sur une courbe où une surface, en insérant plusieurs fois le même nœud ;
2. De subdiviser les courbes ou les surfaces pour en obtenir une restriction ;
3. D'ajouter des points de contrôle pour parvenir à un meilleur contrôle lors du design.

Soit  $\vec{C}^w(u) = \sum_{i=0}^n N_{i,p} \vec{P}_i^w$  une courbe *NURBS* définie sur le vecteur nodal  $U = \{u_0, \dots, u_m\}$ . Soit  $\bar{u} \in [u_k, u_{k+1})$  la valeur du noeud à insérer dans le vecteur nodal  $U$ . On cherche donc à obtenir un nouveau vecteur nodal  $\bar{U}$  qui a pour valeur  $\bar{U} = \{\bar{u}_0 = u_0, \dots, \bar{u}_k = u_k, \bar{u}_{k+1} = \bar{u}, \bar{u}_{k+2} = \bar{u}_{k+1}, \dots, \}$

Soit  $\nu_U$  l'espace vectoriel d'une fonction *NURBS* définie sur le vecteur nodal  $U$  et soit  $\nu_{\bar{U}}$  l'espace vectoriel d'une autre fonction *NURBS* définie sur le vecteur nodal  $\bar{U}$ , on peut alors affirmer que  $\nu_{\bar{U}}$  est un sous ensemble de  $\nu_U$  :  $\nu_{\bar{U}} \subset \nu_U$ . Par conséquent on peut représenter la courbe  $\vec{C}^w(u)$  avec le vecteur nodal  $\bar{U}$  sous la forme :

$$\vec{C}^w(u) = \sum_{i=0}^{n+1} \bar{N}_{i,p} \vec{Q}_i^w.$$

La méthode d'insertion de nœud dans un vecteur nodal consiste à déterminer les valeurs  $\vec{Q}_i^w$ . L'insertion d'un nœud dans un vecteur nodal est simplement un changement de base vectorielle. Par conséquent, la courbe ou la surface résultante après l'insertion d'un nœud dans le vecteur nodal est géométriquement et paramétriquement inchangée.

Les valeurs  $\vec{Q}_i^w$  peuvent être déterminées en résolvant un système d'équations linéaires. On pose l'égalité suivante :

$$\sum_{i=0}^n N_{i,p} \vec{P}_i^w = \sum_{i=0}^{n+1} \bar{N}_{i,p} \vec{Q}_i^w. \quad (\text{II.2})$$

En substituant  $n+2$  valeurs de  $u$  dans cette équation, on obtient un système linéaire nonsingulier de  $n+2$  équations à  $n+2$  inconnues. Chaque inconnue représentent une valeur recherchée  $\tilde{Q}_i^w$ . Ce système linéaire a la particularité d'être structuré en une bande d'une largeur de  $n+2$  diagonales.

Cependant, on peut résoudre ce système de manière plus efficace en utilisant les propriétés des bases B-splinaires. On sait que pour une base B-splinaire, il y a au plus  $p+1$  fonctions nonnulles sur l'intervalle  $[u_k, u_{k+1})$ . Ainsi, on peut poser l'égalité suivante :

$$\sum_{i=k-p}^k N_{i,p}(u) P_i^w = \sum_{i=k-p}^{k+1} N_{i,p}(u) Q_i^w, \forall u \in [u_k, u_{k+1}),$$

$$\begin{aligned} N_{i,p}(u) &= \bar{N}_{i,p}(u) & i = 0, \dots, k-p-1, \\ N_{i,p}(u) &= \bar{N}_{i+1,p}(u) & i = k+1, \dots, n. \end{aligned}$$

En raison de l'indépendance linéaire des fonctions de base entre elles, on déduit que :

$$\begin{aligned} P_i^w &= Q_i^w & \text{pour } i = 0, \dots, k-p-1, \\ P_i^w &= Q_{i+1}^w & \text{pour } i = k+1, \dots, n. \end{aligned}$$

En utilisant un raisonnement par induction, on peut exprimer une relation entre les fonctions  $N_{i,p}$  et les fonctions  $\bar{N}_{i,p}(u)$  dans le cas  $i = k-p, \dots, k$

$$N_{i,p}(u) = \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+p+1} - \bar{u}_i} \bar{N}_{i,p}(u) + \frac{\bar{u}_{i+p+2} - \bar{u}}{\bar{u}_{i+p+2} - \bar{u}_{i+1}} \bar{N}_{i+1,p}(u). \quad (\text{II.3})$$

En substituant l'équation II.3 dans l'équation II.2 on obtient l'égalité suivante :

$$\begin{aligned}
0 &= \bar{N}_{k-p} (Q_{k-p}^w - P_{k-p}^w) \\
&+ \bar{N}_{k-p+1} \left( Q_{k-p+1}^w - \frac{\bar{u} - u_{k-p+1}}{u_{k+1} - u_{k-p+1}} P_{k-p+1}^w - \frac{u_{k+1} - \bar{u}}{u_{k+1} - u_{k-p+1}} P_{k-p}^w \right) \\
&\vdots \\
&+ \bar{N}_k \left( Q_k^w - \frac{\bar{u} - u_k}{u_{k+p} - u_k} P_k^w - \frac{u_{k+p} - \bar{u}}{u_{k+p} - u_k} P_{k-1}^w \right) \\
&+ \bar{N}_{k+1} (Q_{k+1}^w - P_k^w) .
\end{aligned}$$

Pour  $i = k - p + 1, \dots, k$  on pose :

$$\begin{aligned}
\alpha_i &= \frac{\bar{u} - u_i}{u_{i+p} - u_i}, \\
\text{et } 1 - \alpha_i &= \frac{u_{i+p} - \bar{u}}{u_{i+p} - u_i}.
\end{aligned}$$

En substituant et en utilisant l'indépendance linéaire, on a :

$$\begin{aligned}
Q_{k-p}^w &= P_{k-p}^w, \\
Q_i^w &= \alpha_i P_i^w + (1 - \alpha_i) P_{i-1}^w, \quad k - p + 1 \leq i \leq k, \\
Q_{k+1}^w &= P_k^w.
\end{aligned}$$

Finalement on obtient une formulation générale pour l'obtention des nouveaux sommets du polygone de contrôle :

$$Q_i^w = \alpha_i P_i^w + (1 - \alpha_i) P_{i-1}^w,$$

avec

$$\alpha_i = \begin{cases} 1 & i \leq k - p, \\ \frac{\bar{u} - u_i}{u_{i+p} - u_i} & k - p + 1 \leq i \leq k, \\ 0 & i \geq k + 1. \end{cases}$$

On insère un nœud dans une direction paramétrique donnée ( $u$  ou  $v$ ), sur une surface, en appliquant l'algorithme d'insertion de nœuds à chaque ligne ou colonne du polygone de contrôle.

## ANNEXE III

### SURFACES DE BÉZIER RATIONNELLES

Les surfaces de Bézier rationnelles sont un cas particulier des surfaces *NURBS*. Une surface (ou une *patch*) de Bézier de degré  $p$  dans la direction  $u$  et de degré  $q$  dans la direction  $v$  est une *NURBS* définie sur un vecteur nodal ayant la forme suivante :

$$\begin{aligned} U &= \left\{ \underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1} \right\}, \\ V &= \left\{ \underbrace{0, \dots, 0}_{q+1}, \underbrace{1, \dots, 1}_{q+1} \right\}. \end{aligned}$$

Afin de faciliter la manipulation et la caractérisation de surfaces gauches complexes modélisées à l'aide de *NURBS*, il est souvent nécessaire de subdiviser la surface *NURBS* en plusieurs *patches* de Bézier élémentaires.

#### III.1 Polynômes de Bernstein

Les polynômes de Bernstein  $B_{i,n}(t)$  sont définis comme :

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} (1-t)^{n-i} t^i, \quad i = 0, \dots, n.$$

Ces polynômes possèdent des propriétés intéressantes qui permettent de les utiliser comme une base pour la construction de modèles géométriques. Les propriétés les plus importantes sont énumérées ci-dessous :

**Propriété 7 (Non-négativité)**  $B_{i,n}(t) \geq 0, \quad 0 \leq t \leq 1, \quad i = 0, \dots, n.$

**Propriété 8 (Partition de l'unité)**  $\sum_{i=0}^n B_{i,n}(t) = (1-t+t)^n = 1.$

**Propriété 9 (Récursivité)**  $B_{i,n}(t) = (i-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t)$  et  $B_{i,n}(t) = 0$  pour  $i < 0, i > n$  et  $B_{0,0}(t) = 1$ .

**Propriété 10 (Symétrie)**  $B_{i,n}(t) = B_{n-i,n}(1-t)$ .

**Propriété 11 (Élévation du degré)** Un polynôme de Bernstein de degré  $n$  peut être exprimé à l'aide d'un polynôme de Bernstein de degré  $n+1$

$$B_{i,n}(t) = \left(1 - \frac{i}{n+1}\right) B_{i,n+1}(t) + \frac{i+1}{n+1} B_{i+1,n+1}(t).$$

### III.1.1 Formulation des surfaces de Bézier

Une surface de Bézier rationnelle est donnée par le produit tensoriel suivant :

$$\vec{S}^w = \frac{\sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) w_{i,j} \vec{P}_{i,j}^w}{\sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) w_{i,j}}.$$

On peut aussi écrire la surface de Bézier comme :

$$\vec{S} = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) \vec{P}_{i,j},$$

avec

$$R_{i,j}(u, v) = \frac{B_{i,b}(u) B_{j,n}(v) w_{i,j}}{\sum_{r=0}^n \sum_{s=0}^m B_{r,n}(u) B_{s,m}(v) w_{r,s}}.$$

### III.2 Subdivision matricielle des surfaces de Bézier

La subdivision d'une surface de Bézier cubique a été implémentée par multiplication matricielle. On peut exprimer une surface de Bézier bicubique  $P(u, v)$  par :

$$P(u, v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix},$$

où la matrice  $M$  est définie par :

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}.$$

Pour subdiviser la surface en  $u = 0.5$  on peut reparamétriser la formulation précédente afin de couvrir la première moitié de la surface, en effectuant la substitution

$$u = \frac{1}{2}u :$$

$$\begin{aligned}
P\left(\frac{u}{2}, v\right) &= \begin{bmatrix} 1 & \left(\frac{u}{2}\right) & \left(\frac{u}{2}\right)^2 & \left(\frac{u}{2}\right)^3 \end{bmatrix} M \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} M \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M S_L \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}.
\end{aligned}$$

La matrice  $S_L$  est calculée de la façon suivante :

$$\begin{aligned}
S_L &= M^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} M \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{bmatrix}.
\end{aligned}$$

Le polygone de contrôle de la surface couvrant la paramétrisation  $u \in [0, 0.5]$  est

donc donné par :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{bmatrix} \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}.$$

La seconde partie de la surface est obtenue de la même façon. On peut exprimer la deuxième moitié de la surface comme :

$$\begin{aligned} P\left(\frac{1}{2} + \frac{u}{2}, v\right) &= \begin{bmatrix} 1 & (\frac{1}{2} + \frac{u}{2}) & (\frac{1}{2} + \frac{u}{2})^2 & (\frac{1}{2} + \frac{u}{2})^3 \end{bmatrix} M \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \\ &= \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{3}{8} \\ 0 & 0 & \frac{1}{4} & \frac{3}{8} \\ 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} M \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \\ &= \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M S_R \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}, \end{aligned}$$

où

$$S_R = \begin{bmatrix} \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Le polygone de contrôle de la surface couvrant la paramétrisation  $u \in [0.5, 1]$  est

donc donné par :

$$\begin{bmatrix} 0.125 & 0.375 & 0.375 & 0.125 \\ 0 & 0.125 & 0.5 & 0.125 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$$

Il est possible de généraliser ce raisonnement et de subdiviser une surface d'ordre  $n$  pour n'importe quel paramètre  $u, v$ .