



HAL
open science

Design géométrique de surfaces de topologie arbitraire

Riadh Taleb

► **To cite this version:**

Riadh Taleb. Design géométrique de surfaces de topologie arbitraire. Mathématiques [math]. Université Joseph-Fourier - Grenoble I, 2001. Français. NNT: . tel-00004706

HAL Id: tel-00004706

<https://theses.hal.science/tel-00004706v1>

Submitted on 17 Feb 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

Présentée par
Riadh TALEB

Pour obtenir le grade de
Docteur de l'Université Joseph Fourier - Grenoble I

(arrêtés ministériels du 5 juillet 1984 et du 30 mars 1992)

Discipline: **Mathématiques Appliquées**

Design géométrique de surfaces de topologie arbitraire

Soutenue le 01 octobre 2001 devant le jury :

Mme. Annick MONTANVERT	<i>Présidente</i>
M. Marc DANIEL	<i>Rapporteur</i>
Mme. Gudrun ALBRECHT	<i>Rapporteur</i>
M. Bernard LACOLLE	<i>Examineur</i>
Mme. Stefanie HAHMANN	<i>Directeur de thèse</i>

Thèse préparée au sein du Laboratoire de Modélisation et Calcul - Grenoble

Remerciements

Ce travail a été réalisé au sein de l'équipe de Modélisation Géométrique et Approximation du Laboratoire de Modélisation et Calcul de Grenoble sous la responsabilité de Madame S. Hahmann et Monsieur B. Lacolle.

Je tiens à exprimer toute ma gratitude et ma reconnaissance à Madame Stefanie Hahmann, Maître de conférence à l'INP de Grenoble et directeur de cette thèse, qui a su me guider, me soutenir et me conseiller tout au long de ces années de thèse.

Je remercie Monsieur G.-P. Bonneau, professeur au CNRS, qui a énormément contribué par sa compétence, son aide et sa disponibilité, sans lesquelles ce manuscrit n'aurait jamais vu le jour.

J'adresse également mes remerciements à Madame Annick Montanvert, professeur à l'UPMF (Grenoble 2) pour m'avoir fait l'honneur de présider le jury de cette thèse.

Je souhaite exprimer ma gratitude à Monsieur Marc Daniel et à Madame Gudrun Albrecht pour m'avoir fait l'honneur de rapporter sur cette thèse. Je vous remercie pour l'intérêt que vous avez porté à ce travail et pour vos précieux conseils et remarques.

Je remercie énormément Monsieur Bernard Lacolle, professeur à l'université Joseph Fourier pour sa sympathie, sa gentillesse et sa générosité et pour avoir accepté de faire partie du jury.

Je n'oublierai pas les thésards du LMC qui ont contribué à une ambiance au labo, et surtout : Alexandre, Valérie, Khaled, François, et Guillaume.

Mes remerciements s'adressent aussi à toute la famille Rajhi pour sa gentillesse et son soutien dans des moments difficiles.

Je remercie ma mère, mon père, mon frère Fouzi et mes soeurs qui m'ont toujours soutenu et encouragé durant mes années d'études.

Enfin, un grand merci à ma femme Hind, qui a toujours cru en moi, même durant les moments les plus difficiles.

Résumé

Cette thèse est consacrée à la définition d'une surface géométriquement lisse interpolant un ensemble triangulé de points de \mathbb{R}^3 . Une telle triangulation, que nous appelons "réseau surfacique", doit définir une sous-variété de dimension 2, et peut représenter des surfaces de n'importe quel genre topologique. Il fournit l'information topologique, par l'intermédiaire d'une structure de données contenant les informations d'adjacence entre les sommets, les arêtes et les faces. Nous avons développé deux méthodes pour l'interpolation des sommets du réseau surfacique. Elles sont strictement locales et produisent des surfaces polynomiales par morceaux de degré 5 et de continuité G^1 . De nombreux paramètres libres sont disponibles et ajustés soit interactivement soit automatiquement afin de lisser la surface. Dans le contexte interactif, plusieurs outils de design sont développés, basés sur l'interprétation géométrique des paramètres libres. La forme voulue peut être obtenue par une modélisation temps réel, grâce à la localité des algorithmes. Dans le cas du design automatique, de nombreux algorithmes ont été développés satisfaisant un certain nombre de caractéristiques de forme. Un grand nombre de règles heuristiques et d'optimisations locales sont utilisées pour définir les valeurs des paramètres de forme dans le but d'obtenir des formes satisfaisantes ainsi qu'un contrôle optimal de la surface.

Mots-clé : continuité géométrique, surface, patch de Bézier, maillages surfaciques, twist, interpolation locale dans \mathbb{R}^3 , minimisation fonctionnelle, modélisation géométrique, interpolation spline, paramètres de forme.

Abstract

This thesis is concerned with the definition of geometrically smooth surface interpolating a triangulated set of points in \mathbb{R}^3 . Such a triangulation, called a *surface mesh* in our context, is 2-manifold in the 3D space and can represent surfaces of arbitrary topological genus. It gives the topological information by means of a data structure containing the adjacency informations between the vertices, the edges and the faces. We have developed two methods for interpolating the vertices of the surface mesh. They are completely local and produce piecewise polynomial surfaces of degree 5 with G^1 continuity. A large number of free parameters are available and can be adjusted, either interactively or automatically, in order to smooth the surface. In the interactive case, a number of design handles are developed, based on the geometric interpretations of the free parameters. The desired form can be designed in real time thanks to the localness of the algorithms. In the case of automatic design, many algorithms have been developed to satisfy a number of shape features. A large number of simple heuristics and local optimizations are used to determine the shape parameters, that gives pleasing shapes and an optimal control over the surface.

Keywords : geometric continuity, surface, Bézier patch, surface meshes, twist, local interpolation in \mathbb{R}^3 , smooth surface design, functional minimization, geometric modeling, spline interpolation, shape parameters.

Table des matières

Introduction	1
1 Notions de base	7
1.1 Courbes de Bézier	8
1.1.1 Propriétés	8
1.1.2 Algorithme de De Casteljau	9
1.1.3 Élevation de degré	9
1.2 Facettes triangulaires de Bézier	10
1.2.1 Notations	10
1.2.2 Algorithme de De Casteljau triangulaire	12
1.2.3 Dérivée	13
1.2.4 Tangentes trans-frontière	13
1.2.5 Raccordement C^1 entre deux facettes triangulaires de Bézier	15
1.3 Continuité Géométrique	17
1.3.1 Introduction	17
1.3.2 Raccordement G^1 entre deux surfaces	18
1.3.3 Continuité G^1 entre deux facettes triangulaires	19
2 Interpolants G^1	25
2.1 Introduction	25
2.2 Notations	26
2.3 Le problème d'interpolation	27
2.4 Problème de twist	29
2.5 Approche de subdivision	31
2.5.1 Méthode de Piper	31
2.5.2 Méthode de Jensen	33
2.6 Approche de combinaisons convexes	33
2.6.1 Méthode de Gregory	34
2.7 Approche de facettes dégénérées	36
2.8 Approche des facettes algébriques	36

2.9	Approche des courbes frontière	37
2.9.1	Méthode de Loop	37
2.9.2	Méthode de Peters	42
2.10	Discussion	46
2.11	Conclusion	47
3	Méthode de 4-split	49
3.1	Introduction	49
3.2	Notations	50
3.3	Traitement du problème de twist	52
3.4	Choix de Loop des fonctions réelles Φ_i, μ_i, ν_i	53
3.4.1	Motivation du choix de Loop	53
3.5	Subdivision du triangle domaine	53
3.6	Choix des fonctions Φ_i, μ_i et ν_i par la méthode du 4-split	54
3.7	Construction du réseau de courbes frontière	59
3.7.1	Construction de $\bar{\mathbf{b}}_0$: Interpolation ou Approximation	61
3.7.2	Construction de $\bar{\mathbf{b}}_1$	61
3.7.3	Construction de $\bar{\mathbf{b}}_2$	62
3.7.4	Construction de $\bar{\mathbf{b}}_3$	63
3.8	Calcul du vecteur twist compatible	63
3.9	Les dérivées trans-frontière	65
3.10	Macro-facettes sous la forme de Bézier	68
3.10.1	Calcul des points de contrôle intérieurs des macro-facettes	68
3.10.2	Construction des points de contrôle inconnus	72
4	Première généralisation de la méthode du 4-split	75
4.1	Le schéma du 4-split n'est pas strictement local	76
4.2	Localisation du schéma du 4-split	78
4.3	Degrés de liberté de la méthode	81
4.3.1	Choix optimal des points \mathbf{b}_1^i	81
4.3.2	Interpolation des vecteurs normaux	82
4.4	Quelques résultats	85
4.5	Paramètres de forme	86
5	Deuxième Généralisation de la méthode 4-split	87
5.1	Introduction	87
5.2	Relecture du problème de twist	88
5.3	Propriétés des matrices circulantes	90
5.3.1	Application 1 : propriété de la matrice \mathbf{M}_t	92
5.3.2	Application 2 : propriétés de la matrice \mathbf{M}_r	93

5.3.3	Caractérisation de l'espace $Im(\mathbf{M}_t)$	95
5.3.4	Caractérisation de l'espace $ker(\mathbf{M}_r)$	96
5.4	Deuxième généralisation de la méthode	98
5.4.1	Notation	99
5.4.2	Nouveau calcul des dérivées premières	99
5.4.3	Détermination pratique des vecteurs libres a et b	100
5.4.4	Nouveau calcul des dérivées secondes	101
5.4.5	Détermination pratique des dérivées secondes	102
5.4.6	Vecteurs twists compatibles	103
5.4.7	Calcul des dérivées trans-frontière	103
5.4.8	Remplissage des macro-facettes	105
5.4.9	Cas particuliers	105
5.4.10	Conclusion	106
6	Paramètres de forme et leur choix	109
6.1	Design interactif	110
6.2	Paramètres de forme de la première généralisation	112
6.2.1	choix du paramètre β	113
6.2.2	Discussion et résultats	115
6.2.3	Choix du voisinage virtuel: optimisation de forme	115
6.2.4	Choix du voisinage virtuel: interpolation du vecteur normal	116
6.2.5	Choix des paramètres γ_1, γ_2	116
6.2.6	Méthodes heuristiques pour la construction du point et de la tangente cible	118
6.2.7	Discussion et résultats	120
6.2.8	Choix des points de contrôle intérieurs	121
6.2.9	Première approche: choisir les trois points internes de S^m par une technique d'élevation de degré	124
6.2.10	Deuxième approche: les six points internes de S^m sont déterminés par la minimisation d'une énergie	129
6.2.11	Résultats et discussion	130
6.3	Paramètres de forme de la deuxième généralisation	131
6.3.1	Choix de la base du plan tangent	131
6.3.2	Choix optimal des degrés de liberté des dérivées secondes	132
6.3.3	Résultats et discussion	133
6.4	Traitement des réseaux surfaciques ouverts	137
6.4.1	Description de l'algorithme: traitement du bord	137
6.4.2	Complément du voisinage d'un sommet au bord du réseau	138
6.5	Conclusion	140

Conclusion	141
A Calcul des deux premières rangés des points de contrôle d'un patch	145
A.1 Les points de contrôle de la première (rangée) dérivée de M	146
B Structures des données dans le programme interactif	149
B.1 Structures de données	149
B.2 Format de stockage des réseaux surfaciques : le format NeiGhBour	150
Bibliographie	150

Introduction

Historiquement, la conception géométrique assistée par ordinateur (CGAO) fut l'un des premiers domaines à utiliser les courbes et surfaces splines pour la représentation et la modélisation des objets. Le besoin en courbes et surfaces de formes libres dans l'industrie automobile, navale et aéronautique était à l'origine des premiers modèles mathématiques. P. Bézier [6, 7, 9] chez Renault et P.F. de Casteljaou chez Citroën, introduisaient les courbes et les facettes de Bézier. Coons [24] chez General Motors et Jordan chez Ford, introduisaient les facettes de Coons interpolant un réseau de courbes. Depuis, leur application dans beaucoup de domaines ne cesse de croître.

Construire une surface à partir d'un ensemble de points qui contrôlent intuitivement la forme de la surface grâce à la nature des fonctions de base de Bernstein-Bézier ou B-spline est devenue, en CAGD, la méthode la plus populaire pour la modélisation des surfaces libres (free-form surfaces). Une surface est générée par une carte polynomiale (éventuellement rationnelle) d'un domaine planaire subdivisé en un réseau régulier de rectangles ou de triangles, conduisant à une collection de facettes produit-tensorielles ou triangulaires [15, 43, 76, 111, 3]. Mais, en général, cette définition surfacique ne permet pas la représentation de surfaces de topologie quelconque. En effet, une surface B-spline est une application (déformation) d'un domaine planaire subdivisé en une grille régulière de triangles ou de rectangles. Elle est traitée comme une collection de patches polynomiaux définis sur ces triangles ou rectangles. Ceci mène automatiquement à la notion de la continuité paramétrique d'ordre k (notée C^k). Le raccordement entre les patches est exprimé en fonction des dérivées partielles le long des bords de la grille. C'est précisément la cause principale de l'incapacité des B-spline de représenter les surfaces de topologie arbitraire. Les facettes à n côtés [85] peuvent être utilisées pour le remplissage de trous à n côtés qui peuvent se présenter dans une surface rectangulaire et offrent donc la possibilité à représenter des surfaces fermées générales ou des surfaces de topologie quelconque. Néanmoins, la modélisation de surfaces par des facettes à n côtés impose certaines restrictions sur le réseau de courbes.

La méthode la plus courante pour la définition de surfaces sans restriction sur la topo-

logie consiste à utiliser des facettes triangulaires se raccordant continuellement, chacune d'elle étant définie sur le triangle unité. Cette méthode offre l'avantage de permettre une description uniforme pour toutes les topologies possibles sans aucune restriction sur le nombre de faces qui se joignent en un sommet.

La continuité géométrique [37, 43, 84, 113, 31] entre les patches qui composent la surface assure un raccord lisse (visuellement lisse) de la surface. Cette continuité généralise la notion de continuité paramétrique, ne dépend pas de la paramétrisation des patches, et offre plus de degrés de liberté pour contrôler la forme de la surface. Ces surfaces peuvent être générées par l'interpolation ou l'approximation d'une triangulation de topologie quelconque représentant une sous-variété triangulée de l'espace \mathbb{R}^3 . L'application principale est la représentation des objets lisses par une triangulation très fine ou par une surface polynomiale par morceaux, permettant d'effectuer toutes sortes de calculs pour les utilisations de la CAO/CFAO. Une autre application importante est la reconstruction des surfaces pour la réalité virtuelle dans les applications de CAO/CFAO et dans l'informatique graphique.

Cette thèse est consacrée à la définition d'une surface géométriquement lisse par interpolation d'un ensemble triangulé de points de \mathbb{R}^3 . Une telle triangulation, que nous appelons "réseau surfacique", doit définir une sous-variété de dimension 2, et peut représenter des surfaces de n'importe quel type topologique. Il n'y a aucune restriction sur le degré de ses sommets (i.e. le nombre de facettes qui ont ce sommet en commun). De plus, le réseau surfacique fournit l'information topologique, par l'intermédiaire d'une structure de donnée contenant les informations d'adjacences entre les sommets, les arêtes et les faces. Nous supposons que le réseau surfacique est déjà donné.

Il est connu que le raccordement G^1 d'un nombre pair de patches polynomiaux incidents à un sommet est toujours accompagné d'un problème d'incompatibilité, appelé problème de "twist". Pour résoudre ce problème plusieurs types de méthodes existent, elles feront l'objet du chapitre 2 de cette thèse.

Pendant ma thèse, je me suis penché sur la construction de surfaces de topologie quelconque, la modélisation de leurs formes et la réalisation de plusieurs de leurs propriétés géométriques. Les contributions de ce travail se sont inspirées des travaux de Stefanie Hahmann et Georges-Pierre Bonneau [65] sur la méthode du 4-split. Deux méthodes ont été développées pour l'interpolation G^1 reposant principalement sur cette méthode, ainsi qu'une étude détaillée des degrés de liberté et de leur utilisation pratique pour un contrôle optimal de la forme.

Les propriétés importantes des schémas d'interpolation G^1 triangulaires que nous estimons indispensables sont les suivantes :

- La *localité*: on peut dire qu'un schéma de construction de surfaces est local lorsqu'un changement local du réseau surfacique, en entrée, n'affecte la surface d'interpolation que localement. Il existe des méthodes plus locales que d'autres. La notion de méthodes strictement locales signifie que chaque patch de la surface dépend seulement des données de la face correspondante (i.e. ses sommets et ses normales). Il est à noter que les méthodes résumées et discutées dans Peters [107](1990) et dans Mann et al. [92](1992) ne sont pas strictement locales. Dans notre étude, le concept de localité est très important. Les méthodes d'interpolation présentées ici sont strictement locales permettant ainsi des évaluations rapides en vue d'une modélisation interactive.
- L'*interpolation de position* ou de *position-normale*: l'interpolation des sommets du réseau surfacique est l'une des propriétés importantes d'une méthode de construction. Il est aussi connu que l'interpolation des vecteurs normaux influence considérablement la qualité des surfaces produite par la méthode. Une bonne estimation des vecteurs normaux est donc une question importante (pour un résumé des méthodes utilisées pour l'estimation du vecteur normal, voir Sloan 1992 [126]). Dans quelques applications, il est parfois important de pouvoir relaxer l'interpolation à une approximation seulement du réseau surfacique. Il est donc préférable que la méthode de construction puisse basculer entre l'approximation et l'interpolation.
- La *représentation polynomiale explicite* et un degré bas des patches : chaque patch est donné par un petit nombre de points de contrôle, ceci permettra un stockage optimisé de la surface. La représentation explicite par un petit nombre de points de contrôle définissant exactement la surface permet, grâce à quelques routines simples, l'évaluation de beaucoup de caractéristiques surfaciques, comme les dérivées, les vecteurs normaux, les courbures, les lignes de courbures, les isophotes, etc... Ces calculs sont très importants pour certaines applications. Par exemple lors du rendu de la surface sur un écran par un logiciel graphique, l'affichage est obtenu en attribuant une couleur à chaque pixel de l'écran. Cette couleur dépend de plusieurs paramètres : le point projeté de la surface sur ce pixel, la position de la lumière et les propriétés matérielles de la surface. Ces paramètres nécessitent des calculs sur la surface. Une surface sous sa forme polynomiale explicite avec un bas degré facilite beaucoup ces calculs. On peut facilement produire la couleur, les propriétés matérielles déterminant comment la surface reflète la lumière et même les coordonnées de texture, jusqu'à l'obtention d'une image réelle de la surface (pour plus de détail, voir [131]). On peut aussi afficher la surface sous forme d'un réseau de courbes iso-paramétriques (wire frame) afin de voir les parties éventuellement cachées dans le cas solide.
- *Invariance affine* du schéma : Le rendu d'une surface dans un logiciel graphique 3D

a pour but de fournir une image 2D de la surface (pour un affichage sur écran par exemple). Pour cela, on utilise toujours plusieurs transformations affines comme la translation, la rotation, l'homothétie ou une composition d'un ensemble de ces transformations. En général, lorsqu'on veut transformer une surface, on transforme chacun de ses points par un produit matriciel avec une matrice 3×3 . Si la surface est dessinée par des milliers de points, ces opérations risquent de prendre un temps considérable de calcul. Mais si la surface est sous la représentation de Bézier alors elle est invariante par transformation affine, et il suffit de transformer les points de contrôle de la surface.

- Les *paramètres de forme* offerts par la méthode permettent de satisfaire directement ou automatiquement certaines propriétés géométriques, ou encore de contrôler localement la forme de la surface.

Les constructions proposées dans cette thèse sont guidées par ces critères et produisent des surfaces de qualité permettant de satisfaire beaucoup de contraintes de forme.

La thèse est constituée de 6 chapitres. Le chapitre 1 est consacré à des rappels préliminaires contenant les concepts fondamentaux des courbes et surfaces de Bézier, de la continuité paramétrique, de la continuité géométrique dans un cas plus général puis une caractérisation plus détaillée de la continuité géométrique entre les facettes de Bézier triangulaires sera donnée dans sect. 1.3.3.

Le chapitre 2 présente, d'une manière générale, le problème de twist. Un état de l'art des méthodes d'interpolation G^1 résumant les solutions déjà existantes du problème de twist. Une discussion de ces méthodes, basée sur leur utilisation en pratique, terminera la partie introductive de cette thèse.

Le chapitre 3 sera consacré à une étude détaillée de la méthode du 4-split [65], parce qu'elle est à l'origine des travaux de la présente thèse. Cependant, à plusieurs endroits, nous présentons des preuves théoriques de certaines propriétés, contribuant ainsi à la version originale de cette méthode.

Le chapitre 4 propose un premier algorithme généralisant la méthode du 4-split. Elle devient strictement locale et permet de satisfaire quelques propriétés géométriques importantes de la surface comme l'interpolation du plan tangent.

Le chapitre 5 présente une deuxième généralisation de la méthode du 4-split. La méthode résultante sera strictement locale et riche de plusieurs paramètres permettant un meilleur contrôle de forme de la surface. Une nouvelle manière, plus générale, de résoudre le problème de twist sera détaillée.

Le chapitre 6 est consacré à l'utilisation pratique des paramètres de forme offerts par les trois méthodes (la méthode de base et les 2 généralisations). Des méthodes heuristiques et d'autres basées sur l'optimisation de forme pour le choix automatique de ces paramètres seront aussi exposées. Nous montrerons de nombreux résultats numériques qui illustrent

l'effet des différents paramètres de forme et nous discuterons, de façon critique, la capacité des schémas d'interpolation développés dans cette thèse, ainsi que le choix proposé des paramètres associés.

Chapitre 1

Notions de base

Dans ce chapitre, nous allons présenter les bases mathématiques des courbes et surfaces triangulaires de Bézier, sur lesquelles les chapitres suivants vont s'appuyer. Les courbes et les surfaces de Bézier constituent le standard idéal pour la représentation des courbes et surfaces polynomiales par morceaux de formes libres. Farouki et Rajan [44, 45] ont démontré que la base de Bernstein pour les courbes de Bézier est la base la plus stable du point de vue numérique de toutes les autres bases polynomiales connues jusqu'ici. Les courbes et surfaces de forme libre ont été introduites dans les années soixante par P. Bézier [6, 7, 9] chez Renault, P.F. de Casteljaou chez Citroën, Coons [24] chez General Motors et d'autres. Depuis, leur utilisation dans la modélisation de surfaces dans un environnement de conception assistée par ordinateur ne cesse de croître. Dans l'industrie aéronautique, automobile et navale, la construction et l'édition de courbes et surfaces (comme celles des tôles de carrosserie) à l'aide des systèmes de CAO/CFAO a connu beaucoup de progrès, car la forme des modèles tend, de plus en plus, à coïncider avec les courbes et surfaces tracées. Au fil des années, et avec la révolution de l'industrie des ordinateurs, les courbes et surfaces de forme libre trouvent leur place dans plusieurs domaines allant de la modélisation des tissus et des organes en imagerie médicale jusqu'au contrôle de la qualité de production en industrie de textile et de chaussures, en passant par la synthèse d'images en réalité virtuelle pour l'industrie des films, des jeux vidéo et pour la publicité.

Les courbes et surfaces que nous allons considérer dans cette thèse sont les courbes et surfaces triangulaires de Bézier. Dans les sections suivantes, nous allons brièvement rappeler les définitions, quelques propriétés ainsi que les conditions de raccordement d'ordre un. Pour plus de détails, les démonstrations des propositions, ainsi que des applications, nous renvoyons aux ouvrages standards ou spécialisés qui donnent une liste plus complète des références à ce sujet : Hoschek et Lasser [76], Farin [43], [42], [38], [41], Böhm et al. [15], Faux et Pratt [47], Hoffmann [73], Léon [81], Risler [115], Foley et al. [49], Watt [130].

1.1 Courbes de Bézier

Soit $\{\mathbf{b}_i \in \mathbb{R}^d, i = 0, \dots, n\}$ ($d = 2, 3$ dans ce document) une suite ordonnée de $n + 1$ points, appelés points de contrôle ou points de Bézier. Un segment de Bézier est un arc de courbe paramétrique polynomiale de degré n définie par les points de contrôle de la manière suivante

$$\mathbf{x}(t) := \sum_{i=0}^n \mathbf{b}_i B_i^n(t), \quad t \in [a, b],$$

où $B_i^n(t) = \binom{n}{i} \left(\frac{t-a}{b-a}\right)^i \left(\frac{b-t}{b-a}\right)^{n-i}$ sont les polynômes de Bernstein de degré n . Ces polynômes de Bernstein forment une base de \mathcal{P}_n , l'espace vectoriel des polynômes de degré $\leq n$. Un des avantages de la représentation d'un polynôme dans la base de Bernstein se trouve dans la signification géométrique de ses coefficients, ici les \mathbf{b}_i . Ces points forment un polygone qui est approché par le segment de courbe. Il décrit grossièrement la forme de la courbe, d'où son nom polygone de contrôle.

Sans perte de généralité, nous allons dans la suite prendre l'intervalle $[a, b] = [0, 1]$.

Toutes les propriétés des courbes de Bézier se dérivent directement des propriétés des polynômes de Bernstein :

Formule de récurrence :

$$B_i^n(t) = \begin{cases} (1-t)B_i^{n-1}(t) & i=0 \\ (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) & i=1, \dots, n-1 \\ tB_{i-1}^{n-1}(t) & i=n \end{cases} \quad (1.1)$$

Valeurs aux extrémités : $B_i^n(0) = \delta_{i,0}$, $B_i^n(1) = \delta_{i,n}$, $0 \leq i \leq n$

Dérivée :

$$(B_i^n)'(t) = \begin{cases} -nB_i^{n-1}(t) & i=0 \\ n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)) & i=1, \dots, n-1 \\ nB_{i-1}^{n-1}(t) & i=n \end{cases}$$

Partition de l'unité : $\sum_{i=0}^n B_i^n(t) \equiv 1, \forall t \in [0, 1]$

Positivité : $B_i^n(t) \geq 0, \forall t \in [0, 1], 0 \leq i \leq n$.

Produit :

$$B_i^m(t)B_i^n(t) = \frac{C_m^i C_n^j}{C_{m+n}^{i+j}} B_{i+j}^{m+n}(t)$$

1.1.1 Propriétés

Les propriétés les plus importantes des courbes de Bézier sont :

Interpolation :

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{b}_0 \text{ et } \mathbf{x}(1) = \mathbf{b}_n. \\ \mathbf{x}'(0) &= n(\mathbf{b}_1 - \mathbf{b}_0) \text{ et } \mathbf{x}'(1) = n(\mathbf{b}_n - \mathbf{b}_{n-1}). \end{aligned}$$

Enveloppe convexe : la courbe Bézier se trouve dans l'enveloppe convexe de son polygone de contrôle. Ces deux propriétés justifient le nom de "polygone de contrôle" pour la suite des points de contrôle.

Invariance affine : Les courbes de Bézier sont invariantes par transformation affine.

Invariance par transformation affine du paramètre : la courbe de Bézier est définie sur un intervalle $[a, b]$, mais on peut toujours changer cet intervalle en un autre quelconque et la courbe de Bézier restera inchangée. C'est pour cette raison qu'une courbe de Bézier est très souvent définie sur $[0, 1]$.

Voir [76] pour plus de détails.

1.1.2 Algorithme de De Casteljau

L'évaluation de la courbe en un paramètre t peut être effectuée avec l'algorithme de De Casteljau. Il consiste en une suite de combinaisons affines des points de contrôle et peut être représenté par un schéma triangulaire comme illustré en fig. (1.1).

Algorithme:

$$\mathbf{b}_i^0 := \mathbf{b}_i, \quad i = 0, \dots, n$$

Pour $k = 1, \dots, n$, on calcule les $n - k + 1$ points intermédiaires

$$\mathbf{b}_i^k(t) = (1 - t)\mathbf{b}_i^{k-1}(t) + t\mathbf{b}_{i+1}^{k-1}(t) \quad i = 0, \dots, n - k.$$

Le point sur la courbe correspondant au paramètre t est $\mathbf{x}(t) = \mathbf{b}_0^n$.

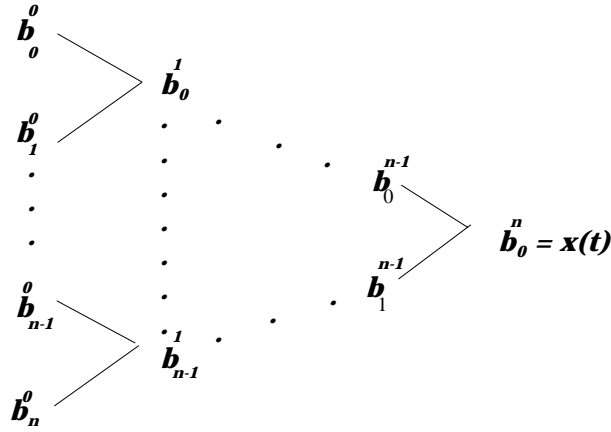
1.1.3 Élévation de degré

L'élévation de degré d'une courbe de Bézier revient à accroître d'une unité le nombre de sommets du polygone de contrôle sans changer la courbe: les nouveaux sommets du polygone de contrôle sont les $n + 2$ points $\tilde{\mathbf{b}}_i$, pour $i := 0, \dots, n + 1$ définis par

$$\begin{cases} \tilde{\mathbf{b}}_0 := \mathbf{b}_0, & \tilde{\mathbf{b}}_{n+1} := \mathbf{b}_n, \\ \tilde{\mathbf{b}}_i := (1 - \frac{i}{n+1})\mathbf{b}_i + \frac{i}{n+1}\mathbf{b}_{i-1}; & i = 1, \dots, n. \end{cases} \quad (1.2)$$

Les nouveaux points de contrôle vérifient

$$x(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t) = \sum_{i=0}^{n+1} \tilde{\mathbf{b}}_i B_i^{n+1}(t) \quad \text{pour } t \in [0, 1].$$

FIG. 1.1 – *Algorithme de De Casteljau*

Donc la courbe reste inchangée. Il faut noter que le nouveau polygone est encore dans l'enveloppe convexe des points de contrôle et approche la courbe mieux que le polygone initial (voir [43, 76]).

1.2 Facettes triangulaires de Bézier

Historiquement, les facettes triangulaires de Bézier ont été introduites d'abord par De Casteljau en 1959 comme une extension naturelle des courbes de Bézier. Le travail de De Casteljau n'a jamais été publié mais celui de Bézier l'a été. C'est pour cette raison que ces facettes portent son nom¹.

Les facettes triangulaires de Bézier sont des patches polynomiaux à trois coins. Elles sont bien adaptées pour représenter la surface d'un objet de topologie complexe, là où les surfaces de type produit tensoriel sont très difficilement utilisables.

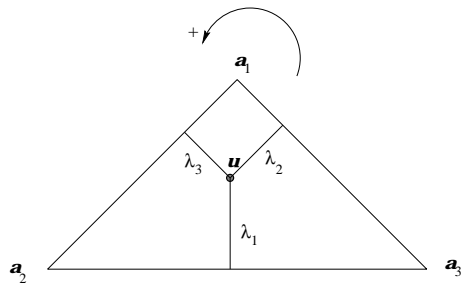
Dans cette section, nous rappelons les définitions et quelques propriétés des facettes triangulaires de Bézier définies sur un domaine triangulaire.

1.2.1 Notations

Soit Δ un triangle de \mathbb{R}^2 de sommets \mathbf{a}_i , $i = 1, 2, 3$. Les coordonnées barycentriques d'un point $\mathbf{u} = (u, v) \in \Delta$ sont définies par les trois nombres réels λ_i , $i = 1, 2, 3$, avec λ_i la distance perpendiculaire de \mathbf{u} à l'arête opposée au sommet \mathbf{a}_i , fig. 1.2. Nous avons donc

$$\begin{cases} \mathbf{u} = \lambda_1 \mathbf{a}_1 + \lambda_2 \mathbf{a}_2 + \lambda_3 \mathbf{a}_3, \\ 1 = \lambda_1 + \lambda_2 + \lambda_3. \end{cases} \quad (1.3)$$

¹ G. Farin appelait la plupart des résultats relatifs à ce sujet avec le nom de Bézier

FIG. 1.2 – Coordonnées barycentriques d'un point $\mathbf{u} = (u, v) \in \Delta$

Les coordonnées barycentriques λ_i , $i = 1, 2, 3$ de \mathbf{u} sont des fonctions de ses coordonnées euclidiennes (u, v) . Mais les coordonnées barycentriques représentent mieux le domaine triangulaire que les coordonnées euclidiennes.

On définit aussi la paramétrisation des arêtes de Δ par

$$E_i(t) = (1 - t)\mathbf{a}_i + t\mathbf{a}_{i+1}, \quad t \in [0, 1], \quad i = 1, 2, 3 \text{ (modulo 3)}.$$

L'arête orientée E_i est donc le segment allant de \mathbf{a}_i à \mathbf{a}_{i+1} , elle correspond à $\lambda_{i+2} = 0$. Les indices sur les sommets sont pris modulo 3.

Une facette de Bézier triangulaire M est définie sur Δ par

$$M(\mathbf{u}) = \sum_{\mathbf{i} \in c_n} \mathbf{b}_i B_{\mathbf{i}}^n(\lambda_1, \lambda_2, \lambda_3), \quad \mathbf{u} = (\lambda_1, \lambda_2, \lambda_3) \in \Delta, \quad (1.4)$$

où $\mathbf{i} = (i, j, k)$ est un triple-indice parcourant l'ensemble

$$c_n = \{(i, j, k) \in \mathbb{N}^3 : i + j + k = n\},$$

contenant $\frac{1}{2}(n+1)(n+2)$ triplets et $B_{\mathbf{i}}^n$ sont les polynômes de Bernstein généralisés, définies dans [15] par

$$B_{\mathbf{i}}^n(\lambda_1, \lambda_2, \lambda_3) = \frac{n!}{i!j!k!} \lambda_1^i \lambda_2^j \lambda_3^k. \quad (1.5)$$

Les points \mathbf{b}_i sont appelés les points de contrôle de la facette de Bézier M , la figure 1.3 montre les points de contrôle d'une facette de Bézier quartique.

L'image par M de l'arête $E_i = [\mathbf{a}_i \mathbf{a}_{i+1}]$ est une courbe de Bézier de degré n , ses points de contrôle correspondent aux points frontière du réseau de contrôle de la facette.

Remarque 1 *L'orientation du triangle est donnée par l'ordre des sommets. On suppose que la direction positive est le sens inverse des aiguilles d'une montre. Ainsi les paramétrisations des arêtes $E_i(t)$, $i = 1, 2, 3$ sont telles que t croît dans ce sens.*

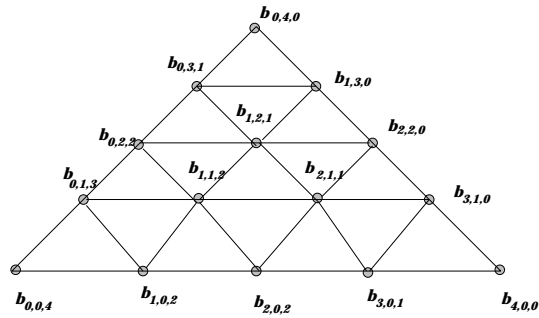


FIG. 1.3 – Réseau de contrôle d'une facette triangulaire quartique

1.2.2 Algorithme de De Casteljau triangulaire

L'algorithme de De Casteljau pour les facettes triangulaires de Bézier est une généralisation naturelle de l'algorithme pour les courbes, ce qui explique pourquoi les facettes triangulaires de Bézier sont la prolongation naturelle des courbes de Bézier.

Algorithme de De Casteljau :

- On part du réseau triangulaire $\mathbf{b}_i^0 = \mathbf{b}_i$, $|\mathbf{i}| = n$ à $\frac{1}{2}(n+2)(n+1)$ points et d'un point dans le domaine triangulaire (u, v) de coordonnées barycentriques $(\lambda_1, \lambda_2, \lambda_3)$.
- On construit les $\frac{1}{2}(n+1)n$ points $\mathbf{b}_i^1 = \lambda_1 \mathbf{b}_{i+\mathbf{e}_1}^0 + \lambda_2 \mathbf{b}_{i+\mathbf{e}_2}^0 + \lambda_3 \mathbf{b}_{i+\mathbf{e}_3}^0$, pour $|\mathbf{i}| = n-1$, avec $\mathbf{e}_1 = (1, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0)$ et $\mathbf{e}_3 = (0, 0, 1)$. Les points \mathbf{b}_i^1 forment à leur tour un nouveau réseau triangulaire.
- A partir du réseau \mathbf{b}_i^1 , $|\mathbf{i}| = n-1$, on recommence l'opération précédente, et on obtient $\frac{1}{2}n(n-1)$ points \mathbf{b}_i^2 , $|\mathbf{i}| = n-2$.
- Après n itérations, on obtient un point $b_{(0,0,0)}^n$, qui est le point recherché appartenant à la facette triangulaire, i.e. $M(u, v) = \mathbf{b}_{(0,0,0)}^n$. Fig. 1.4.

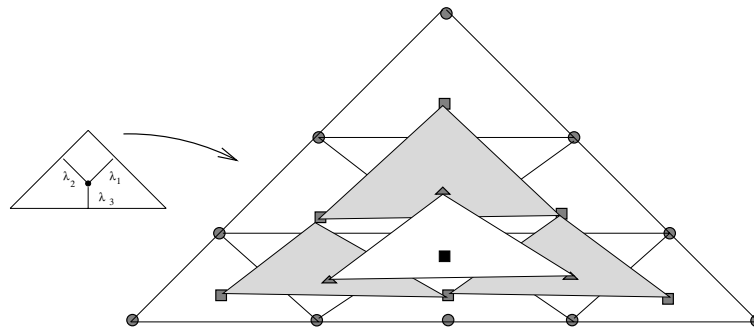


FIG. 1.4 – Schéma de l'algorithme De Casteljau triangulaire

Remarque 2 *L'algorithme utilise des combinaisons affines répétées, expliquant à la fois la propriété de l'invariance affine et la propriété de l'enveloppe convexe des facettes triangulaires de Bézier.*

Remarque 3 *Les trois points $\mathbf{b}_{(1,0,0)}^{n-1}$, $\mathbf{b}_{(0,1,0)}^{n-1}$, $\mathbf{b}_{(0,0,1)}^{n-1}$, obtenus à l'étape $n - 1$ de l'algorithme de De Casteljau définissent le plan tangent en $M(\mathbf{u})$.*

1.2.3 Dérivée

Les dérivées directionnelles d'un patch de Bézier triangulaire sont plus appropriées et très souvent utilisées.

Soient \mathbf{m}_1 et \mathbf{m}_2 deux points distincts du triangle Δ , leurs coordonnées barycentriques sont respectivement (ρ_1, ρ_2, ρ_3) et $(\sigma_1, \sigma_2, \sigma_3)$. Considérons le vecteur $\mathbf{v} = \mathbf{m}_2 - \mathbf{m}_1$, de coordonnées barycentriques données par $(\tau_1, \tau_2, \tau_3) = (\sigma_1 - \rho_1, \sigma_2 - \rho_2, \sigma_3 - \rho_3)$, elles vérifient $\tau_1 + \tau_2 + \tau_3 = 0$.

La dérivée directionnelle de la surface M en un point $\mathbf{u} = (\lambda_1, \lambda_2, \lambda_3)$ suivant la direction du vecteur \mathbf{v} est donnée par (voir Farin [43]):

$$D_{\mathbf{v}}M(\mathbf{u}) = \sum_{\mathbf{i} \in c_{n-1}} B_{\mathbf{i}}^{n-1}(\lambda_1, \lambda_2, \lambda_3) [\tau_1 \mathbf{b}_{\mathbf{i}+\mathbf{e}_1} + \tau_2 \mathbf{b}_{\mathbf{i}+\mathbf{e}_2} + \tau_3 \mathbf{b}_{\mathbf{i}+\mathbf{e}_3}], \quad (1.6)$$

où $\mathbf{e}_1 = (1, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0)$, $\mathbf{e}_3 = (0, 0, 1)$, et $D_{\mathbf{v}}$ est l'opérateur de dérivation suivant la direction \mathbf{v} . La formule (1.6) nous permet de définir les vecteurs tangents en tout point sur la facette triangulaire de Bézier $M(\mathbf{u})$. Le plan tangent en \mathbf{u} est défini par deux dérivées directionnelles dans deux directions linéairement indépendantes dans le triangle domaine.

On peut aussi définir les dérivées directionnelles supérieures d'une facette triangulaire de Bézier par :

$$D_{\mathbf{v}}^r M(\mathbf{u}) = \frac{n!}{(n-r)!} \sum_{\mathbf{i} \in c_{n-r}} \mathbf{d}_{\mathbf{i}}^r B_{\mathbf{i}}^{n-r}(\lambda_1, \lambda_2, \lambda_3). \quad (1.7)$$

Les vecteurs $\mathbf{d}_{\mathbf{i}}^r$, $\mathbf{i} \in c_{n-r}$ sont obtenus par la formule de récurrence suivante :

$$\mathbf{d}_{\mathbf{i}}^r = \tau_1 \mathbf{d}_{\mathbf{i}+\mathbf{e}_1}^{r-1} + \tau_2 \mathbf{d}_{\mathbf{i}+\mathbf{e}_2}^{r-1} + \tau_3 \mathbf{d}_{\mathbf{i}+\mathbf{e}_3}^{r-1}, \quad (1.8)$$

avec $\mathbf{d}_{\mathbf{i}}^0 = \mathbf{b}_{\mathbf{i}}$, $\forall \mathbf{i} \in c_n$.

1.2.4 Tangentes trans-frontière

Pour définir le plan tangent en un point quelconque de la courbe frontière de la facette M , correspondant à l'arête E_k , nous avons à connaître deux dérivées directionnelles

dans deux directions linéairement indépendantes quelconques du triangle Δ . On choisit généralement les deux directions suivantes :

- la direction \mathbf{u}_k de l'arête E_k allant de \mathbf{a}_k à \mathbf{a}_{k+1} , i.e.

$$\mathbf{u}_k = \mathbf{a}_{k+1} - \mathbf{a}_k,$$

- la direction $-\mathbf{u}_{k+2}$, allant de \mathbf{a}_k à \mathbf{a}_{k+2} , i.e.

$$-\mathbf{u}_{k+2} = \mathbf{a}_{k+2} - \mathbf{a}_k.$$

Par exemple, pour $k = 1$: $\mathbf{u}_1 = (0, 1, 0) - (1, 0, 0) = (-1, 1, 0)$ et $-\mathbf{u}_2 = (0, 0, 1) - (1, 0, 0) = (-1, 0, 1)$.

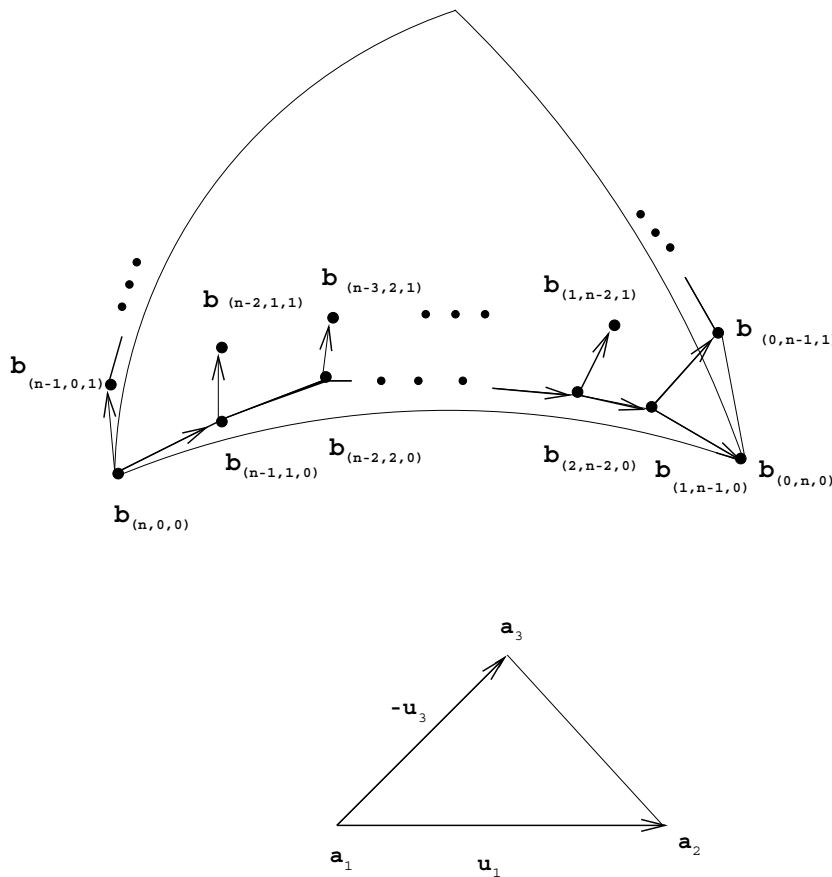


FIG. 1.5 – Interprétation géométrique des deux dérivées $D_{\mathbf{u}_1}M(E_1(t))$ et $D_{-\mathbf{u}_3}M(E_1(t))$.

Définition 1 La dérivée trans-frontière de la facette M , le long de la courbe frontière correspondant à l'arête E_k est la dérivée directionnelle de M dans la direction $-\mathbf{u}_{k+2} = \mathbf{a}_{k+2} -$

\mathbf{a}_k , i.e.

$$[D_{-\mathbf{u}_{k+2}}] M(E_k(t)) = n \sum_{i=0}^{n-1} [\mathbf{b}_{(n-i)\mathbf{e}_k+i\mathbf{e}_{k+1}+\mathbf{e}_{k+2}} - \mathbf{b}_{(n-i)\mathbf{e}_k+i\mathbf{e}_{k+1}}] B_i^{n-1}(t), \quad t \in [0, 1]. \tag{1.9}$$

Pour $k = 1$, voir fig. 1.5, la dérivée trans-frontière le long de la courbe $M(E_1(t))$ est un “ruban” de vecteurs défini sous forme de Bézier, le premier vecteur de contrôle étant $n(\mathbf{b}_{(n-1,0,1)} - \mathbf{b}_{(n,0,0)})$ et le dernier vecteur de contrôle étant $n(\mathbf{b}_{(0,n-1,1)} - \mathbf{b}_{(0,n,0)})$. Le premier vecteur est la tangente de la courbe frontière de M correspondant à l’arête $M(E_3(t))$.

Les tangentes trans-frontière le long d’une courbe dépendent seulement des points de Bézier de cette courbe et de la première rangée intérieure des points de contrôle de la facette triangulaire, qui sont directement adjacents aux points de la courbe.

Ce résultat est essentiel pour les raccordements C^1 et G^1 entre deux facettes triangulaires ayant une courbe frontière commune.

1.2.5 Raccordement C^1 entre deux facettes triangulaires de Bézier

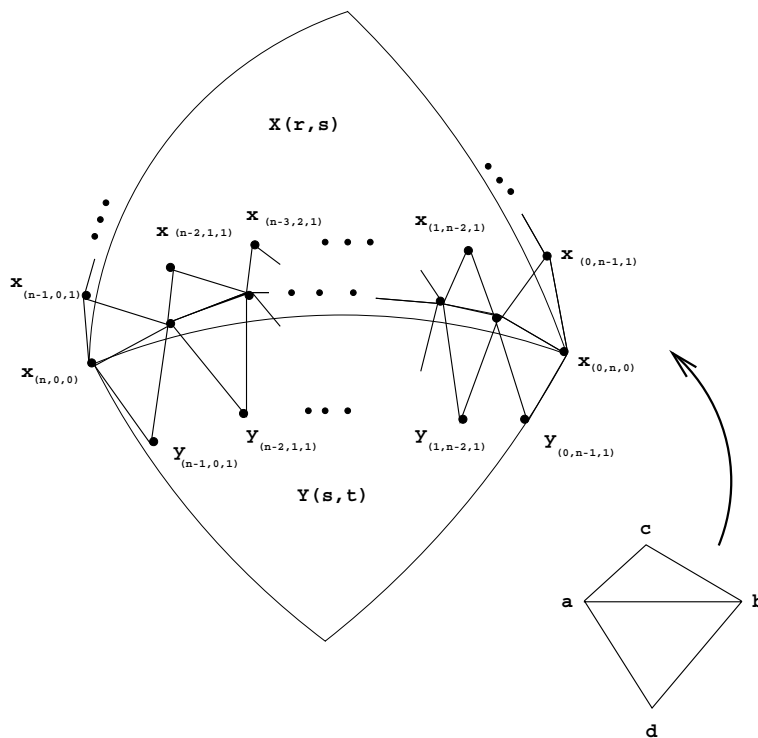


FIG. 1.6 – Conditions de raccordement C^1 entre deux facettes triangulaires de Bézier.

Soient $X(r, s)$ et $Y(s, t)$ deux facettes triangulaires de Bézier de degré n , adjacentes et C^0 continues, i.e.

$$X(0, s) = Y(s, 0), \forall s \in [0, 1].$$

$X(r, s)$ est définie sur le triangle $\Delta_1 = \mathbf{abc}$ et $Y(s, t)$ sur le triangle adjacent $\Delta_2 = \mathbf{abd}$, voir fig. 1.6. Considérons maintenant la surface Z définie sur le quadrilatère \mathbf{acbd} composée des deux facettes X et Y . La surface Z est C^∞ sur le quadrilatère \mathbf{acbd} sauf à travers l'arête $[\mathbf{ab}]$, où elle est seulement C^0 .

Quelles sont les conditions sur les points de contrôle de X et Y pour que la facette composite Z soit C^1 ?

Considérons les nombres $\lambda_1, \lambda_2, \lambda_3$ les distances signées entre le point \mathbf{d} et les droites (\mathbf{bc}) , (\mathbf{ca}) et (\mathbf{ab}) respectivement.

On dit que $X(r, s)$ et $Y(s, t)$ se raccordent C^1 à travers leur courbe commune si la surface Z est C^1 sur \mathbf{acbd} et ceci est vérifiée si et seulement si : (voir [43] et fig. 1.6)

$$\mathbf{y}_{(n-1-j,j,1)} = \lambda_1 \mathbf{x}_{(n-j,j,0)} + \lambda_2 \mathbf{x}_{(n-j-1,j+1,0)} + \lambda_3 \mathbf{x}_{(n-1-j,j,1)}, \quad j = 0, \dots, n-1, \quad (1.10)$$

où $\mathbf{x}_{(i,j,k)}$ et $\mathbf{y}_{(i,j,k)}$ sont respectivement les points de contrôle de X et Y .

Chacun des n points de contrôle de Y , situés directement près de la courbe commune est obtenu par la même combinaison linéaire **barycentrique**¹ de trois points de contrôle de X (deux points sur la première rangée commune et un point sur la deuxième rangée).

Géométriquement, les conditions 1.10 signifient que chaque paire de triangles montrée en figure 1.6 est une carte affine des deux triangles Δ_1 et Δ_2 .

Pour plus de détails voir [43], pages 289-293.

Remarque si les deux triangles Δ_1 et Δ_2 sont équilatéraux, alors (1.10) peut être interprétée géométriquement par les paires de triangles montrées figure 1.7 qui deviennent des parallélogrammes, car chaque paire est la carte affine du parallélogramme composé des deux triangles Δ_1 et Δ_2 .

Nous traiterons plus tard les surfaces composées d'un réseau de facettes triangulaires formant une seule surface lisse. Concentrons-nous d'abord sur une autre notion de continuité entre deux facettes adjacentes, celle de la continuité géométrique.

1. Cette combinaison n'est pas convexe, car \mathbf{d} n'est pas à l'intérieur de Δ_1 .

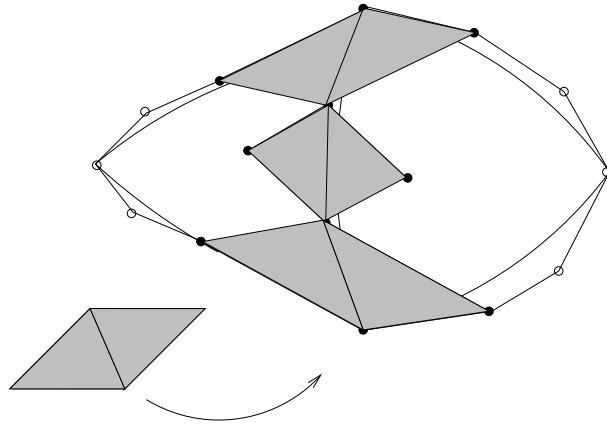


FIG. 1.7 – Conditions de parallélogrammes.

1.3 Continuité Géométrique

1.3.1 Introduction

Le raccordement C^1 entre les facettes surfaciques dépend de la paramétrisation des facettes (voir la section précédente). Ainsi les surfaces fermées² en particulier et les surfaces de topologie arbitraire en général ne peuvent pas être représentées par un raccordement C^1 de facettes [72, 85]. D'autre part, ces conditions C^1 sont trop fortes et géométriquement peu significatives. On peut relâcher la condition de continuité C^1 pour la continuité géométrique de premier ordre. Deux facettes surfaciques de courbe frontière commune sont dites G^1 continues si le plan tangent est continu le long de la courbe commune. Ceci implique qu'en chaque point de la courbe frontière commune, toutes les dérivées premières sont co-planaires.

Cette définition est la généralisation de la continuité C^1 : toutes les surfaces C^1 sont G^1 , l'inverse n'est pas vraie. Des contre-exemples ont été donnés dans [43] et [72].

La G^1 continuité est géométriquement plus significative et ne dépend pas de la paramétrisation, ce qui la rend plus adaptée à la représentation des surfaces de forme libre. C'est pour ces raisons que la construction des surfaces G^1 continues est un problème important en CAGD.

Dans la suite, nous allons traiter la G^1 continuité entre deux facettes uniquement. Les résultats de cette étude seront exploités dans les chapitres suivants pour le raccordement d'un nombre arbitraire de facettes qui partagent un seul sommet.

2. une surface fermée est une surface sans bord.

1.3.2 Raccordement G^1 entre deux surfaces

Considérons $Q(u, v)$ et $R(v, w)$ deux paramétrisations C^1 de deux facettes surfaciques Q et R , qui partagent une même courbe commune C de paramétrisation $C(v) = Q(0, v) = R(v, 0)$, voir fig. 1.8.

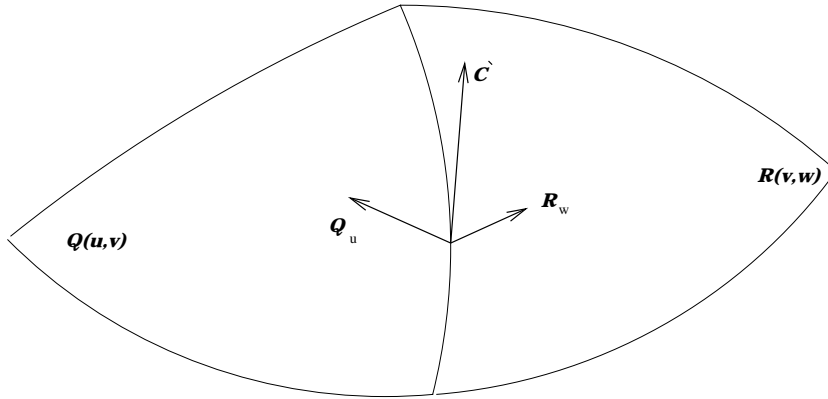


FIG. 1.8 – Continuité G^1 entre deux facettes.

Définition 2 Q et R sont dites G^1 continues si et seulement si les conditions suivantes sont satisfaites : [135, 110, 124, 133, 134, 113, 84]

– Continuité G^0 entre Q et R , i.e.

$$C(v) = Q(0, v) = R(v, 0), \quad (1.11)$$

– Il existe trois fonctions réelles $\Phi(v)$, $\mu(v)$ et $\nu(v)$ telles que

$$\Phi(v)C'(v) = \mu(v) \left. \frac{\partial Q}{\partial u} \right|_C + \nu(v) \left. \frac{\partial R}{\partial w} \right|_C, \quad (1.12)$$

où $\left. \frac{\partial Q}{\partial u} \right|_C$ (resp. $\left. \frac{\partial R}{\partial w} \right|_C$) est la dérivée trans-frontière de la facette Q (resp. de la facette R) le long de la courbe frontière C .

– Une orientation correcte de chacune des surfaces par rapport à l'autre :

$$\mu(v)\nu(v) > 0, \quad (1.13)$$

– Plan tangent bien défini :

$$\left. \frac{\partial Q}{\partial u} \right|_C(0, v) \times \left. \frac{\partial Q}{\partial v} \right|_C(0, v) \neq 0. \quad (1.14)$$

La condition 1.13 est pour préserver l'orientation des deux surfaces, l'une par rapport à l'autre. Ceci est pour éviter les plis et les auto-intersections de la surface, voir aussi [84],

page 307 et [110], page 225.

La condition 1.12 est nécessaire et suffisante pour la co-planarité des vecteurs $\frac{\partial Q}{\partial u}, \frac{\partial R}{\partial w}$ et C' mais d'autres conditions nécessaires et suffisantes pouvant la remplacer existent.

Par exemple les deux conditions 1.15 et 1.16 suivantes :

$$\det \left(\frac{\partial Q}{\partial u}(0, v), \frac{\partial R}{\partial w}(v, 0), C'(v) \right) = 0, \quad \forall v \in [0, 1]. \quad (1.15)$$

(voir [124, 125] pour plus de détail).

– Ou l'existence de 4 fonctions réelles $p_1(v), q_1(v), p_2(v)$ et $q_2(v)$ et d'une fonction vectorielle $\mathbf{V}(v)$ telles que

$$\begin{cases} \frac{\partial Q}{\partial u}|_C = p_1(v)\mathbf{V}(v) + q_1(v)C'(v), \\ \frac{\partial R}{\partial w}|_C = p_2(v)\mathbf{V}(v) + q_2(v)C'(v). \end{cases} \quad (1.16)$$

(voir [135]).

On remarque que lorsque les deux surfaces Q et R sont polynomiales, toutes les fonctions intermédiaires seront polynomiales (voir [110]).

La condition 1.15 ne peut pas être utilisée dans la construction de facettes G^1 , car elle mène à un système d'équations non-linéaires, puisqu'elle mélange les coordonnées des trois vecteurs suivants les trois axes de l'espace euclidien \mathbb{R}^3 .

Les deux équations 1.12 et 1.16 sont deux manières équivalentes d'exprimer la dépendance linéaire des trois vecteurs $\frac{\partial Q}{\partial u}, \frac{\partial R}{\partial w}$ et C' et de la continuité du plan tangent le long de C . Dans la section suivante, nous allons traiter un cas particulier qui nous intéresse dans la suite : la continuité géométrique de deux facettes triangulaires de Bézier.

1.3.3 Continuité G^1 entre deux facettes triangulaires

Les conditions de la G^1 continuité entre deux facettes de Bézier triangulaires ont été étudiées par plusieurs auteurs (voir par exemple [37], [43], [84], [113] [31]). Considérons deux facettes de Bézier adjacentes Q et R de degré n . La facette Q (resp. R) est paramétrée par $(u, v) \in \Delta_1 = \mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$ (resp. $(v, w) \in \Delta_2 = \mathbf{a}_1\mathbf{a}_2\tilde{\mathbf{a}}_3$). La courbe frontière correspondant à l'arête $[\mathbf{a}_1\mathbf{a}_2]$ est commune entre Q et R , voir figure 1.9. Comme dans [113, 124, 92, 43] et afin d'alléger l'écriture, notons par $\mathbf{c}_i, i = 0, \dots, n$ les points de contrôle de cette courbe et par $\mathbf{q}_i, \mathbf{r}_i, i = 0, \dots, n - 1$ les points de contrôle des facettes Q, R situés directement près des points \mathbf{c}_i . Posons

$$\mathbf{u}_1 = \mathbf{a}_2 - \mathbf{a}_1, \quad \mathbf{u}_2 = \mathbf{a}_3 - \mathbf{a}_1, \quad \tilde{\mathbf{u}}_2 = \tilde{\mathbf{a}}_3 - \mathbf{a}_1. \quad (1.17)$$

Avec ces notations, les deux facettes Q et R se raccordent G^1 si et seulement s'il existe des fonctions réelles $\Phi(v), \mu(v)$ et $\nu(v)$ telles que

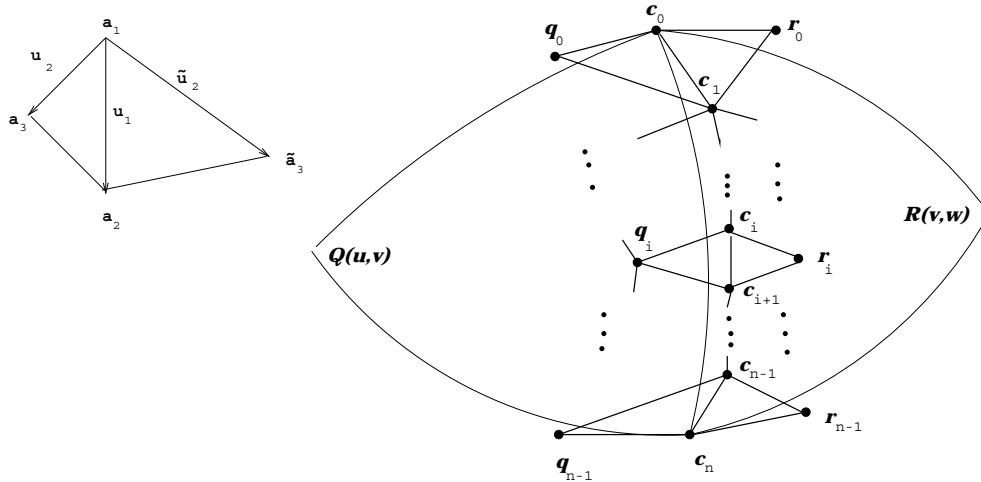


FIG. 1.9 – Notations.

$$\Phi(v)Q_{\mathbf{u}_1}(0, v) = \mu(v)Q_{\mathbf{u}_2}(0, v) + \nu(v)R_{\tilde{\mathbf{u}}_2}(v, 0),$$

$$Q_{\mathbf{u}_1} \times Q_{\mathbf{u}_2} \neq 0, \tag{1.18}$$

$$\mu\nu > 0.$$

Il est à noter que $Q_{\mathbf{u}_2}$ et $R_{\tilde{\mathbf{u}}_2}$ sont les dérivées trans-frontière de Q et R . Elles ne dépendent que des points de contrôle de la courbe commune et des deux rangées de points de contrôle à gauche et à droite de cette courbe. $Q_{\mathbf{u}_1}$ étant la dérivée de la courbe. Ce qui nous laisse dire, comme pour le raccordement de C^1 continuité, que la continuité G^1 des deux facettes Q et R dépend seulement de ces 3 rangées de points de contrôle, voir fig.1.9.

Les conditions 1.18 sont une caractérisation algébrique des conditions nécessaires et suffisantes pour la continuité G^1 . Dans les paragraphes suivants, nous exposerons deux approches différentes pour la construction d'une surface G^1 composée de deux facettes triangulaires de Bézier.

Approche de Farin

L'approche de Farin [37] consiste à prendre μ et ν (les facteurs de $Q_{\mathbf{u}_2}(0, v)$ et $R_{\tilde{\mathbf{u}}_2}(v, 0)$ dans 1.18) constantes et de prendre $\Phi(v)$ linéaire, ce qui laisse 3 coefficients scalaires indéterminés, car on peut toujours prendre l'un des coefficients μ ou ν égal à 1. Cependant, les conditions de la co-planarité aux deux extrémités de la courbe commune donnent 4

contraintes. C'est pour cette raison que Farin pose une contrainte supplémentaire sur les points de contrôle des courbes frontière, exprimée en termes des aires de triangles hachurés de la figure 1.10 : (voir aussi [124], page 283)

$$\frac{\overline{\text{aire}}(\mathbf{r}_0 \mathbf{c}_0 \mathbf{c}_1)}{\overline{\text{aire}}(\mathbf{q}_0 \mathbf{c}_0 \mathbf{c}_1)} = \frac{\overline{\text{aire}}(\mathbf{r}_{n-1} \mathbf{c}_{n-1} \mathbf{c}_n)}{\overline{\text{aire}}(\mathbf{q}_n \mathbf{c}_{n-1} \mathbf{c}_n)}. \quad (1.19)$$

Les contraintes de la co-planarité et la dernière contrainte peuvent s'écrire algébriquement, en termes des coordonnées barycentriques, sous la forme :

$$\begin{cases} \mathbf{r}_0 = \alpha_1 \mathbf{c}_0 + \alpha_2 \mathbf{c}_1 + \alpha \mathbf{q}_0, & \alpha_1 + \alpha_2 + \alpha = 1, \\ \mathbf{r}_n = \alpha_3 \mathbf{c}_{n-1} + \alpha_4 \mathbf{c}_n + \alpha \mathbf{q}_n, & \alpha_3 + \alpha_4 + \alpha = 1, \end{cases} \quad (1.20)$$

qui expriment \mathbf{r}_0 et \mathbf{r}_n par rapport aux coordonnées barycentriques des deux triangles $\mathbf{c}_0 \mathbf{c}_1 \mathbf{q}_0$ et $\mathbf{c}_{n-1} \mathbf{c}_n \mathbf{q}_{n-1}$ afin d'assurer que les points \mathbf{r}_0 , \mathbf{c}_0 , \mathbf{c}_1 et \mathbf{q}_0 sont co-planaires ainsi que les points \mathbf{r}_n , \mathbf{c}_{n-1} , \mathbf{c}_n et \mathbf{q}_n . Tandis que $\alpha_1 + \alpha_2 = \alpha_3 + \alpha_4$ est équivalente à la contrainte 1.19.

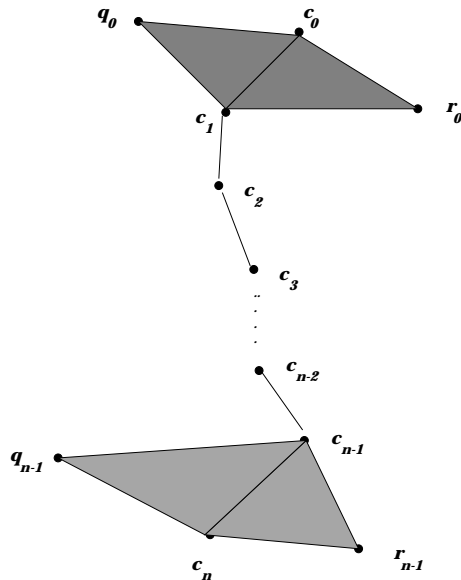


FIG. 1.10 – Construction de Farin de deux facettes G^1 .

La continuité géométrique entre Q et R est ensuite assurée par les contraintes linéaires :

$$\mathbf{r}_i = \frac{n-i}{n}(\alpha_1 \mathbf{c}_i + \alpha_2 \mathbf{c}_{i+1} + \alpha \mathbf{q}_i) + \frac{i}{n}(\alpha_3 \mathbf{c}_{i-1} + \alpha_4 \mathbf{c}_i + \alpha \mathbf{q}_i), \quad i = 1, \dots, n-1, \quad (1.21)$$

Le lecteur est invité à comparer les conditions 1.21 avec les conditions 1.10 de la continuité C^1 de la section 1.2.5. Les conditions 1.21 sont moins fortes que les conditions de la C^1

continuité.

La méthode de Farin permet donc de calculer géométriquement les points de Bézier intérieurs qui assurent une continuité G^1 entre les deux facettes. Elle commence par calculer les 5 coefficients $\alpha, \alpha_1, \alpha_2, \alpha_3, \alpha_4$ à partir des équations 1.20, ensuite les points de contrôle intérieurs des deux facettes, immédiatement près de la courbe commune, sont calculés sous les conditions 1.21.

Remarque 4 Les cinq coefficients $\alpha_j, \alpha, j = 1, 2, 3, 4$ ont une relation indirecte avec les coefficients des fonctions Φ, μ et ν .

Approche de Chiyokura-Kimura

Considérons deux facettes triangulaires de Bézier voisines $Q(u, v)$ et $R(v, w)$. Supposons d'abord qu'elles sont G^0 continues, i.e. elles partagent les mêmes points de contrôle le long de leur frontière commune. Ensuite, supposons que leurs courbes frontière sont toutes cubiques et connues (cf. fig. 1.11). La méthode de Chiyokura et Kimura [22] propose une

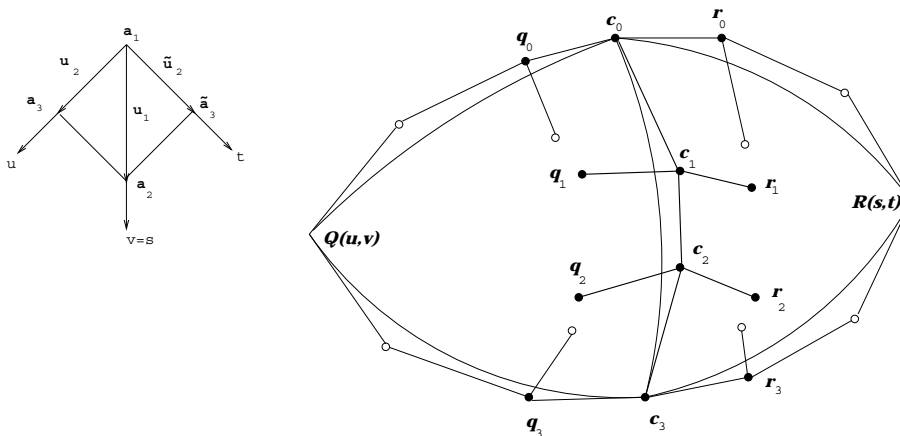


FIG. 1.11 – Conditions G^1 de Chiyokura et Kimura sur les points de contrôle de deux facettes.

autre manière pour construire les points intérieurs $\mathbf{q}_1, \mathbf{q}_2$ et $\mathbf{r}_1, \mathbf{r}_2$ pour que Q et R se raccordent avec G^1 continuité. D'abord elle crée une fonction vectorielle linéaire, qui sert à engendrer les tangentes trans-frontière le long du bord commun :

$$\mathbf{V}(v) = (1 - v)\mathbf{v}_0 + v\mathbf{v}_1,$$

avec

- \mathbf{v}_0 est (à un signe près) le vecteur unitaire perpendiculaire à $\mathbf{c}_1 - \mathbf{c}_0$ dans le plan tangent commun à Q et R au point \mathbf{c}_0 ,

- \mathbf{v}_1 est, de même, le vecteur unitaire perpendiculaire au vecteur $\mathbf{c}_2 - \mathbf{c}_3$ dans le plan tangent commun à Q et R au point \mathbf{c}_3 (cf. fig. 1.11).

Les tangentes trans-frontière de Q et de R le long du bord commun (voir section 1.2.4) sont imposées respectivement par :

$$\begin{aligned} Q_{\mathbf{u}_2}(0, v) &= k(v)\mathbf{V}(v) + h(v)Q_{\mathbf{u}_1}(0, v), \\ R_{\mathbf{u}_2}(v, 0) &= l(v)\mathbf{V}(v) + m(v)Q_{\mathbf{u}_1}(0, v), \end{aligned} \quad v \in [0, 1], \quad (1.22)$$

où $k(v), h(v), l(v), m(v)$ sont des fonctions linéaires. Il est intéressant de noter que la détermination de la dérivée trans-frontière de Q est indépendante de celle de R . Ceci est important pour une détermination locale d'une surface G^1 continue. Nous pouvons alors exposer seulement la construction de $Q_{\mathbf{u}_2}(0, v)$. Écrivons d'abord $k(v) = (1 - v)k_0 + vk_1$ et $h(v) = (1 - v)h_0 + vh_1$. Les constantes k_0, k_1, h_0, h_1 sont calculées en évaluant 1.22 en $v = 0$ et en $v = 1$. Ensuite, afin de trouver les deux points \mathbf{q}_1 et \mathbf{q}_2 , on écrit les deux membres de la première équation de 1.22 dans la base de Bernstein et on compare les coefficients de $B_1^n(t)$ et $B_2^n(t)$, on obtient (voir [92]) :

$$\begin{aligned} \mathbf{q}_1 &= \mathbf{c}_1 + \frac{1}{3}[(k_0 + k_1)\mathbf{v}_0 + k_0\mathbf{v}_1 + 2h_0\mathbf{s}_1 + h_1\mathbf{s}_0], \\ \mathbf{q}_2 &= \mathbf{c}_2 + \frac{1}{3}[k_1\mathbf{v}_0 + (k_0 + k_1)\mathbf{v}_1 + h_0\mathbf{s}_2 + 2h_1\mathbf{s}_1], \end{aligned} \quad (1.23)$$

avec $\mathbf{s}_i = \mathbf{c}_{i+1} - \mathbf{c}_i$, $i = 0, 1, 2$.

Les deux points \mathbf{r}_1 et \mathbf{r}_2 s'obtiennent de manière analogue.

Remarques :

1. La dérivée trans-frontière de Q , donnée par l'équation 1.22 est quadratique. Si la facette Q est construite sous forme d'une facette quartique, alors nous pouvons soit augmenter le degré de $k(t)$ et gagner une liberté scalaire, soit élever le degré de $\mathbf{V}(t)$ et gagner une liberté vectorielle. Ces deux options ont été reprises par Jensen [78] et Herron [72] (voir aussi [89]).
2. Il faut aussi noter qu'au départ, Chiyokura et Kimura ont conçu cette méthode pour les patches quadrilatères de Gregory, mais Shirman et Séquin [124] ont été les premiers à démontrer qu'on peut utiliser cette méthode pour les facettes triangulaires de Bézier.
3. Si on applique cette méthode sur les autres courbes frontière voisines au bord commun, on va certainement produire d'autres points de contrôle intérieurs différents des points déjà obtenus, voir fig. 1.11. Ce problème est appelé problème de compatibilité de twist et il sera traité en détail dans le chapitre suivant.

Dans le chapitre suivant, nous présenterons plusieurs approches existantes pour la construction des surfaces de topologie arbitraire.

Chapitre 2

Interpolants G^1

2.1 Introduction

Ce chapitre est consacré à l'étude de quelques méthodes permettant de construire une surface visuellement lisse, composée de facettes triangulaires, par interpolation d'un ensemble de points de \mathbb{R}^3 , muni d'une triangulation constituant une sous-variété de dimension 2 de \mathbb{R}^3 , appelée "réseau surfacique". Aucune restriction sur le nombre d'arêtes communes à un sommet ne sera imposée. La surface qui interpole un tel réseau se compose de facettes triangulaires en correspondance bijective avec les faces de la triangulation. La topologie (toutes les relations d'adjacences) de cette triangulation reflète ainsi les caractéristiques topologiques de la surface finale. Les sommets de ce réseau sont éventuellement munis de vecteurs normaux.

Il est à noter qu'il existe un domaine de recherche voisin qui s'appelle modélisation ou interpolation de "scattered data". Les données à interpoler sont de la forme $(x_i, y_i; F_i)$, $i = 1, \dots, N$ avec $x_i, y_i, F_i \in \mathbb{R}$. Il s'agit ici de trouver une fonction à valeurs réelles $F : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, telle que $F(x_i, y_i) = F_i$ (analogue pour les dimensions supérieures à 2). Il peut y avoir une triangulation planaire des données sous-jacentes, comme dans le cas du Minimum Norm Network (Nielson [100, 103, 102]), mais ce n'est pas nécessaire. Il existe une littérature abondante sur ce sujet. Une des premières publications remonte aux années 70 (Hardy [70]) et ce sujet continue à l'heure actuel à intéresser les chercheurs. Mais puisque ce n'est pas le sujet de cette thèse, nous nous contentons de ne citer que des références contenant des synthèses comme Alfeld [2], Franke [51] et Franke, Nielson [52] qui donnent un aperçu assez complet des méthodes relatives aux "scattered data".

Bien que le problème qui sera traité dans cette thèse soit parfois appelé dans la littérature "scattered data 3D", nous préférons de ne pas utiliser ce terme pour les raisons suivantes : Il peut y avoir confusion avec les scattered data classique. De plus, il ne s'agit pas vraiment

de données arbitrairement distribuées, car elles sont représentées par des sous-variétés triangulées de dimension 2 et qui peuvent être de topologie quelconque.

Dans la suite de ce chapitre et après avoir fixé les notations dans la première sous-section, le problème d'interpolation d'un réseau surfacique est posé plus en détail (sect. 2.3). Dans la littérature, la difficulté majeure que l'on rencontre ainsi que des critères de comparaison sont notamment mentionnés. La difficulté que chaque méthode d'interpolation G^1 rencontre résulte en une incompatibilité des vecteurs twist aux sommets de degré pair. La section 2.4 décrit d'une manière générale en quoi consiste ce qu'on appelle "le problème de twist". Ensuite en suivant plusieurs sections 2.5 - 2.9, nous présentons les idées principales des différentes méthodes antérieures, sans donner trop de détails, mais nous insistons à chaque fois sur la façon suivant laquelle le problème de twist est résolu. Ceci est important pour bien connaître le contexte historique de ce qui est présenté plus loin dans cette thèse. Un autre aspect très important, pour ne pas dire le plus important, est la qualité des résultats en vue d'une utilisation dans un système graphique 3D. Une discussion de ces méthodes sous cet aspect pratique suivra en sect. 2.10 avant la conclusion du chapitre.

2.2 Notations

Nous avons en entrée un réseau triangulaire surfacique \mathcal{M} qui est supposé être une sous-variété de \mathbb{R}^3 de dimension 2. \mathcal{M} est un ensemble fini de triangles satisfaisant les conditions de cohérence suivantes :

- chaque arête de la triangulation est partagée par au plus deux triangles (sans bifurcations),
- un sommet peut être commun à un nombre arbitraire de triangles,
- chaque triangle ayant un sommet commun avec un autre triangle partage aussi une arête commune avec ce triangle,
- si un sommet d'un triangle est commun à un autre triangle, alors c'est un sommet de ce triangle,
- aucun triangle n'a une intersection avec l'intérieur d'un autre triangle (pas d'overlap).

Notons l'ensemble des sommets de \mathcal{M} par

$$V = \{\mathbf{v}_l \in \mathbb{R}^3 / l = 1, \dots, N_v\}$$

et l'ensemble des vecteurs normaux aux sommets de \mathcal{M} par

$$N = \{\mathbf{n}_l \in \mathbb{R}^3 / \|\mathbf{n}_l\| = 1, l = 1, \dots, N_v\}.$$

Dans la suite, nous utilisons les notations suivantes :

- Nous appelons *triangle domaine* le triangle de \mathbb{R}^3 défini par

$$\Delta = \{(u, v, w) / u + v + w = 1 \text{ et } u, v, w \geq 0\},$$

avec les sommets $\mathbf{a}_1 = (1, 0, 0)$, $\mathbf{a}_2 = (0, 1, 0)$, $\mathbf{a}_3 = (0, 0, 1)$ et les arêtes orientées $\mathbf{u}_1 = \mathbf{a}_2 - \mathbf{a}_1$, $\mathbf{u}_2 = \mathbf{a}_3 - \mathbf{a}_2$ et $\mathbf{u}_3 = \mathbf{a}_1 - \mathbf{a}_3$.

Les nombres u, v, w peuvent aussi être vus comme les coordonnées barycentriques d'un point quelconque de Δ . Notons que Δ est le 2-simplexe standard de \mathbb{R}^3 (voir [98], page 454).

- Les opérateurs $D_{\mathbf{u}_i}$, $i = 1, 2, 3$ des dérivées directionnelles suivant les directions $\mathbf{u}_i = \mathbf{a}_{i+1} - \mathbf{a}_i$, avec $i = 1, 2, 3$. Les indices sont pris modulo 3.

2.3 Le problème d'interpolation

L'interpolation d'un réseau surfacique \mathcal{M} consiste à trouver une surface paramétrique \mathcal{S} , triangulaire par morceau, interpolant les sommets V et éventuellement l'ensemble N des vecteurs normaux. Les facettes de \mathcal{S} se raccordent généralement avec continuité G^1 , i.e. de plan tangent continu. Chaque facette est l'image du triangle domaine et elle est construite en correspondance à un triangle du réseau de telle sorte que ces coins interpolent les sommets du triangle.

Plusieurs approches existent pour résoudre ce problème. Nous nous intéressons aux méthodes *locales*, où une modification locale d'un sommet du réseau n'entraîne qu'une modification locale de la surface. La forme de la surface peut ainsi être contrôlée localement. Toute méthode d'interpolation globale est à écarter pour deux raisons : chaque modification locale du réseau nécessite de recalculer la surface entière. Si le nombre de sommets est grand, une manipulation interactive devient impossible. De plus, un contrôle local de la forme n'est pas possible.

Dans ce chapitre, nous n'allons pas donner une liste exhaustive de toutes les méthodes qui existent, mais nous nous contentons de présenter les méthodes principales, représentant les différentes approches de résolution de ce problème. Nous pouvons partager ces méthodes en cinq approches principales : l'*approche de subdivision* (Farin [39], Piper [113], Shirmann et Séquin [124]), l'*approche de combinaisons convexes* (Nielson [103], Gregory [58]), l'*approche de facettes implicites* (Bajaj et Ihm [4]), l'*approche de facettes dégénérées* (Peters [109], Bohl et Reif [16], Neamtu et Pfluger [98]) et l'*approche des courbes frontière* (Herron [72], Peters [110], Loop [84]).

Le point principal qui différencie ces cinq approches est la résolution du problème de compatibilité de twist (ou simplement problème de twist) : un problème qui apparaît lors du

raccordement G^1 d'un nombre pair de facettes de classe C^2 autour d'un sommet. Chaque courbe de bord, construite par l'algorithme, porte l'information de la dérivée seconde mixte vers les composantes tangentielles de sa dérivée seconde, comme indique notamment Peters [110]. Autour d'un sommet, cette information provient de plusieurs courbes et crée un système de contraintes appelé problème de compatibilité de twist. Si ce sommet possède un nombre impair de sommets voisins de la triangulation alors ce problème peut être traité facilement. Mais s'il possède un voisinage pair et les patches sont au moins deux fois continuellement différentiables, alors la construction fait face à un système d'équations non inversible.

Si un réseau de courbes frontière et les champs de vecteurs transversaux sont donnés le long de ces courbes, alors chaque courbe autour d'un sommet va imposer indépendamment des autres une dérivée seconde mixte en ce sommet (appelé vecteur twist). Les méthodes de *combinaisons convexes* construisent trois patches pour chaque face triangulaire, chacun répondant à une partie des contraintes. Ensuite une combinaison convexe de ces trois patches donne un patch correspondant à cette face et interpolant les trois courbes de bord ainsi que les dérivées transversales. Les surfaces produites par ces méthodes introduisent des discontinuités des dérivées secondes mixtes aux sommets.

L'approche par subdivision raffine le réseau de départ en introduisant de nouveaux sommets. Le but est de résoudre le problème de twist et de réduire le degré polynomial de l'interpolant sans trop augmenter le nombre total des patches. La subdivision du type Clough-Tocher par insertion d'un sommet au barycentre de chaque face, est le schéma le plus utilisé par les méthodes de subdivision. Bien que les sommets de degré impair deviennent de degré pair, cette approche permet de résoudre le problème de twist. En effet, les courbes frontière nouvellement introduites créent des discontinuités des dérivées secondes de la même façon que les méthodes de combinaison convexe (voir [107]).

Dans [110], un seul patch de degré 5 pour chaque face est construit, mais les courbes frontière doivent être C^2 -compatibles. Dans [106], les patches sont de bas degré mais un grand nombre de patches doivent être créés par rapport au nombre de faces de départ. Dans l'article de Loop et DeRose [85], les réseaux surfaciques irréguliers sont approchés par des patches à n côtés appelés S-patches. Le travail développé dans [84] consiste à traiter le problème de lissage des réseaux triangulaires de topologie arbitraire. Les patches utilisés sont polynomiaux de degré 6 et le problème de twist est résolu avec une technique différente de toutes les techniques précédentes. Cependant, cette méthode ne peut être appliquée qu'à l'approximation. L'interpolation est théoriquement possible mais mène à des oscillations indésirables des courbes frontière. La méthode proposée par Hahmann et Bonneau [65], appelée méthode du 4-split, résout le problème d'interpolation avec des facettes de degré au plus égal à 5 en introduisant une subdivision du triangle domaine par les milieux de ces côtés. Cette subdivision produit quatre sous-facettes pour chaque

face. Elle présente certains avantages par rapport à la subdivision de Clough-Tocher. Une approche différente est exposée dans [4] et résout le problème en utilisant des facettes algébriques implicites, correspondant aux faces du réseau. Sederberg [122] a montré plusieurs surfaces implicites lisses sous forme de Bernstein à trois variables et qui peuvent être manipulées comme des fonctions de Bézier par un réseau de contrôle constitué de tétraèdres.

Une autre manière d'interpoler les réseaux triangulaires de topologie quelconque est l'utilisation des schémas de subdivisions récursifs interpolants, qui produisent une surface visuellement lisse en appliquant récursivement au réseau surfacique un ensemble de règles de subdivision. La référence [77] (pages 46–63) présente un résumé de ces techniques. Le schéma de subdivision de type "butterfly", développé par Dyn, Gregory et Levin dans [32] puis amélioré dans [33] est un bon exemple de ces méthodes. Elle résulte en une surface de plan tangent continu lorsque le réseau est de topologie régulière (tous les sommets doivent être de degré six), voir [87]. Zorin et al. [138] ont proposé un schéma de subdivision, appelé MBS (Modified Butterfly Scheme), qui peut être appliqué aux réseaux de topologies quelconques, voir aussi [136], [137], [11]. Elle retient la simplicité de la subdivision "butterfly" et arrive à produire des surfaces lisses en introduisant de nouvelles règles d'interpolation aux sommets irréguliers. L'inconvénient de ce type de méthodes c'est qu'elles ne donnent pas une représentation explicite des surfaces qu'elles produisent. Elles ne se prêtent donc pas à des calculs sur les surfaces comme les intersections ou l'évaluation des quantités géométriques intrinsèques. De plus, lorsque le réseau surfacique est irrégulier alors ces méthodes ne produisent pas de meilleurs résultats que les méthodes polynomiales.

Plusieurs méthodes ont été étudiées et comparées dans Mann [89]. Nous avons cité dans l'introduction de cette thèse un ensemble de conditions suivant lesquelles nous pouvons juger une méthode bonne ou mauvaise.

Dans les sections suivantes, nous rappelons d'abord le problème de twist, ensuite nous enchaînons directement par la présentation des différentes approches du problème d'interpolation. Nous étudions pour chaque approche, au moins une méthode classique de la littérature. Le but est de donner le développement d'idées lors de la solution de ce problème.

2.4 Problème de twist

À la fin de la section 1.3.3, nous avons vu que le raccordement G^1 de deux patches triangulaires voisins dépend seulement des points de contrôle de la courbe commune et des points de contrôle des deux patches situés directement près de cette courbe. Ces points

permettent aussi de calculer les dérivées trans-frontière de chaque facette.

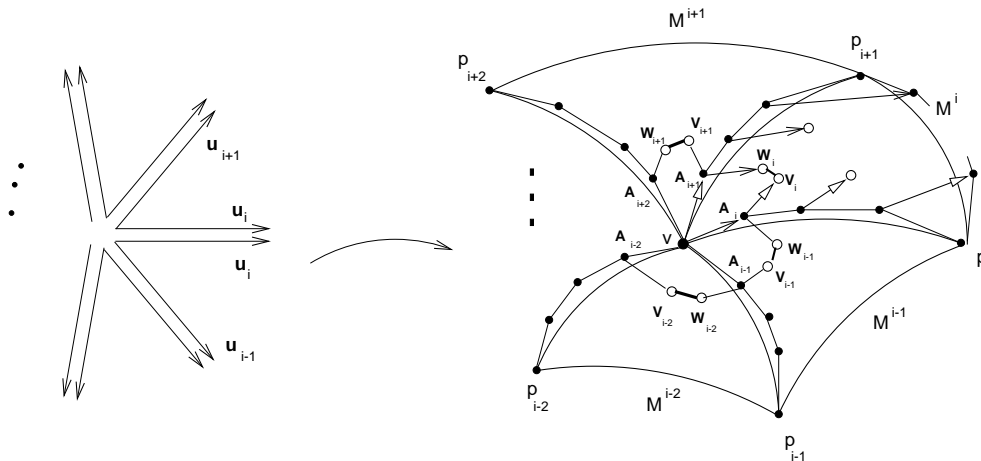


FIG. 2.1 – Problème de twist lors du raccordement de n patches de Bézier de degré 4 ($n \geq 2$).

Considérons à présent le problème du raccordement G^1 de n patches de Bézier M^i , $i = 1, 2, \dots, n$ ayant un sommet commun \mathbf{v} , voir fig. 2.1. Le point \mathbf{v} et les autres sommets \mathbf{p}_i des n patches voisins étant donnés ainsi que les courbes frontière allant de \mathbf{v} aux \mathbf{p}_i . Pour raccorder avec continuité G^1 les n facettes autour du sommet \mathbf{v} un champ de dérivées trans-frontière le long de chaque courbe commune entre deux facettes voisines (M^{i-1}, M^i), ($i = 1, 2, \dots, n$ modulo n) est à construire. Le champ de dérivées trans-frontière entre deux patches voisins peut être construit avec la méthode de Chiyokura-Kimura (sect. 1.3.3) par exemple. Malheureusement, le problème est beaucoup plus complexe s'il s'agit du raccordement de n patches autour d'un sommet, en particulier quand il s'agit de patches polynomiaux. La figure 2.1 montre que lorsqu'on applique cette méthode entre M^{i-1} et M^i et puis entre M^i et M^{i+1} , on construit deux champs de dérivées trans-frontière : $D_{\mathbf{u}_{i+1}} M^i(u_i, 0)$ et $D_{\mathbf{u}_i} M^i(0, u_{i+1})$ le long des deux courbes frontière $\mathbf{v}\mathbf{p}_i$ et $\mathbf{v}\mathbf{p}_{i+1}$ de M^i . La figure 2.1 montre un problème dans la représentation polynomiale des facettes. Les deux points \mathbf{V}_i et \mathbf{W}_i doivent être identiques. Ce problème est connu dans la littérature sous le nom du *problème de twist*.

En effet, le *vecteur twist* est par définition la dérivée deuxième mixte de $M^i(u_i, u_{i+1})$ aux sommets, c'est à dire le vecteur

$$\mathbf{t}_i = [D_{\mathbf{u}_i} D_{\mathbf{u}_{i+1}}] M^i(0, 0).$$

Mais la facette M^i est de classe C^∞ au voisinage du sommet \mathbf{v} . D'où¹

$$\mathbf{t}_i = [D_{\mathbf{u}_{i+1}} D_{\mathbf{u}_i}] M^i(0, 0).$$

Avec la notation de la figure 2.1, on a

$$[D_{\mathbf{u}_i} D_{\mathbf{u}_{i+1}}] M^i(0, 0) = (m - 1)m(\mathbf{v} + \mathbf{V}_i - \mathbf{A}_i - \mathbf{A}_{i+1}), \quad (2.1)$$

et

$$[D_{\mathbf{u}_{i+1}} D_{\mathbf{u}_i}] M^i(0, 0) = (m - 1)m(\mathbf{v} + \mathbf{W}_i - \mathbf{A}_i - \mathbf{A}_{i+1}), \quad (2.2)$$

avec m le degré polynomial des facettes. Ces deux vecteurs sont égaux si et seulement si les deux points \mathbf{V}_i et \mathbf{W}_i sont identiques et définissent donc un seul point, appelé *point twist* de la facette M^i .

En général, ces deux vecteurs n'ont aucune raison d'être égaux. Ceci rend impossible la représentation polynomiale des facettes M^i . C'est le *problème de twist*. La manière de résoudre ce problème est le point principal qui différencie les différentes approches que nous allons présenter.

2.5 Approche de subdivision

Les méthodes de subdivision du domaine par la technique de Clough-Tocher sont parmi les méthodes les plus anciennes ([37], [91], [78]). Elles procèdent de la manière suivante: chaque macro-facette est subdivisée par son barycentre en trois sous-facettes quartiques (voir Farin [39], [40]). Cette subdivision permet une détermination locale des facettes, car elle introduit suffisamment de degrés de liberté afin de satisfaire indépendamment les conditions G^1 le long des courbes frontière adjacentes et résoudre le problème de twist. Les courbes frontière sont des courbes de Bézier cubiques mais élevées en degré.

Prenons comme modèle de cette approche la méthode de Piper [113]. Pour plus de références sur les interpolants qui suivent l'approche de subdivision, consulter les références suivantes: Mann [89], Hansford [69], Peters [106, 108, 107].

2.5.1 Méthode de Piper

Dans la référence [113], Piper a montré que les facettes cubiques ne sont pas toujours capables de résoudre le problème d'interpolation. Et il a proposé un interpolant quartique basé sur une subdivision de type Clough-Tocher, qui peut se résumer en 3 étapes successives:

1. pour chaque face $f \in F$, il construit une facette cubique de la manière suivante:

1. En effet, cette propriété est vérifiée pour toute surface de classe C^2

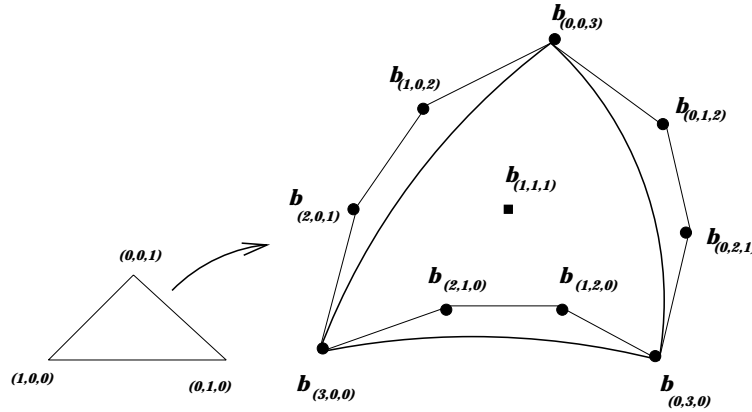


FIG. 2.2 – Paramétrisation de la facette cubique construite dans la première étape.

les trois points de contrôle $\mathbf{b}_{(3,0,0)}$, $\mathbf{b}_{(0,3,0)}$ et $\mathbf{b}_{(0,0,3)}$ sont tout simplement les sommets de f . Les autres points de contrôle frontière sont construits de manière à éviter les éventuelles ondulations des courbes frontière (cf. figure 2.3). Piper obtient $\mathbf{b}_{(2,1,0)}$ en deux étapes : d'abord on projette $\mathbf{b}_{(0,3,0)}$ en un point $\tilde{\mathbf{b}}_{(0,3,0)}$ sur le plan tangent donné en $\mathbf{b}_{(3,0,0)}$ par son vecteur normal. Ceci définit une droite tangente en $\mathbf{b}_{(3,0,0)}$. Puis on place $\mathbf{b}_{(2,1,0)}$ sur cette tangente de la manière suivante :

$$\mathbf{b}_{(2,1,0)} = \mathbf{b}_{(3,0,0)} + \alpha(\tilde{\mathbf{b}}_{(0,3,0)} - \mathbf{b}_{(3,0,0)}).$$

Piper prend $\alpha = \frac{4}{9} \|\tilde{\mathbf{b}}_{(0,3,0)} - \mathbf{b}_{(3,0,0)}\|$. Les autres points $\mathbf{b}_{(1,2,0)}$, $\mathbf{b}_{(0,2,1)}$, $\mathbf{b}_{(0,1,2)}$, $\mathbf{b}_{(1,0,2)}$

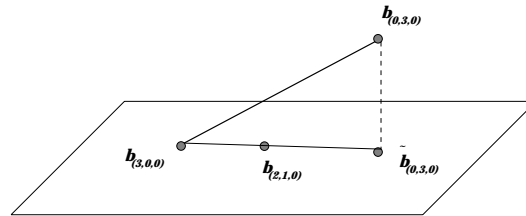


FIG. 2.3 – Construction du point $\mathbf{b}_{(2,1,0)}$ par la méthode de Piper.

et $\mathbf{b}_{(2,0,1)}$ sont obtenus de la même manière.

Cette construction des points frontière assure l'interpolation point-normale. Piper ensuite construit le point central $\mathbf{b}_{(1,1,1)}$ par une combinaison linéaire des points frontière pour avoir une facette de précision linéaire². Ceci est afin d'éviter l'ondulation de la facette cubique.

2. Si l'une des 3 courbes frontière est un segment de droite, alors la facette est réglée.

2. Chaque facette cubique est ensuite subdivisée par son centroïde³ en trois sous-facettes cubiques en utilisant l'algorithme de De Casteljau ([55], sect. 1.2.2). On modifie les points de contrôle intérieurs \mathbf{H} des sous-facettes ayant en commun une courbe frontière initiale (avant la subdivision) afin de rendre les triangles hachurés de la figure 2.4 coplanaires. Ce plan étant le plan moyen pondéré des deux triangles. Dans cette pondération, Piper favorise les triangles de petites hauteurs pour que les facettes ne soient pas trop allongées.
3. La modification de l'étape précédente détruit le raccordement C^1 entre les sous-facettes. Pour maintenir cette continuité, chaque sous-facette est élevée en degré à 4, figure 2.4 (à droite). Les points de contrôle ainsi obtenus sont appelés points candidats. Les points \mathbf{G} offrent un nombre de degrés de liberté dépassant le nécessaire pour assurer un raccordement G^1 . Piper alors les utilise pour satisfaire les conditions G^1 , tout en approchant au mieux les points candidats de l'étape précédente. Les points \mathbf{M} , \mathbf{N} et \mathbf{O} de chaque sous-facette sont modifiés afin de satisfaire les C^1 contraintes (sect. 1.2.5) le long des courbes internes des macro-facettes. Chacun des points \mathbf{M} , \mathbf{N} et \mathbf{O} est placé au centroïde des points du triangle qui l'entoure, voir fig. 2.4 (à droite). Les contraintes C^1 sont alors satisfaites parce que la subdivision de la deuxième étape a été effectuée au centroïde du triangle domaine de la macro-facette (cf. sect. 1.2.5 pour plus de détails).

2.5.2 Méthode de Jensen

La méthode de Jensen [78] ressemble beaucoup à celle de Shirman et Séquin, deux différences les distinguant ont été mentionnées par Mann [89]: d'abord elle utilise une autre version de la méthode de Chiyokura-Kimura pour raccorder deux facettes adjacentes. Dans la section 1.3.3, nous avons mentionné cette version. Ensuite, la méthode de Jensen maintient une continuité C^1 entre les sous-facettes de chaque macro-facette (après subdivision). Tandis que la méthode de Shirman-Séquin utilise un raccordement G^1 entre les sous-facettes voisines.

2.6 Approche de combinaisons convexes

Les méthodes de combinaisons convexes construisent une seule facette pour chaque face f de \mathcal{M} en mélangeant, grâce à des opérateurs d'interpolation, des informations de position et de dérivées connues aux sommets et/ou le long des courbes frontière à l'intérieur de patches. Les surfaces sont en général rationnelles et au moins G^1 continues. Elles ne présentent pas le problème de compatibilité de twist, car le twist est discontinu.

³ le centroïde de la facette est l'image du point aux coordonnées barycentriques $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ du domaine Δ .

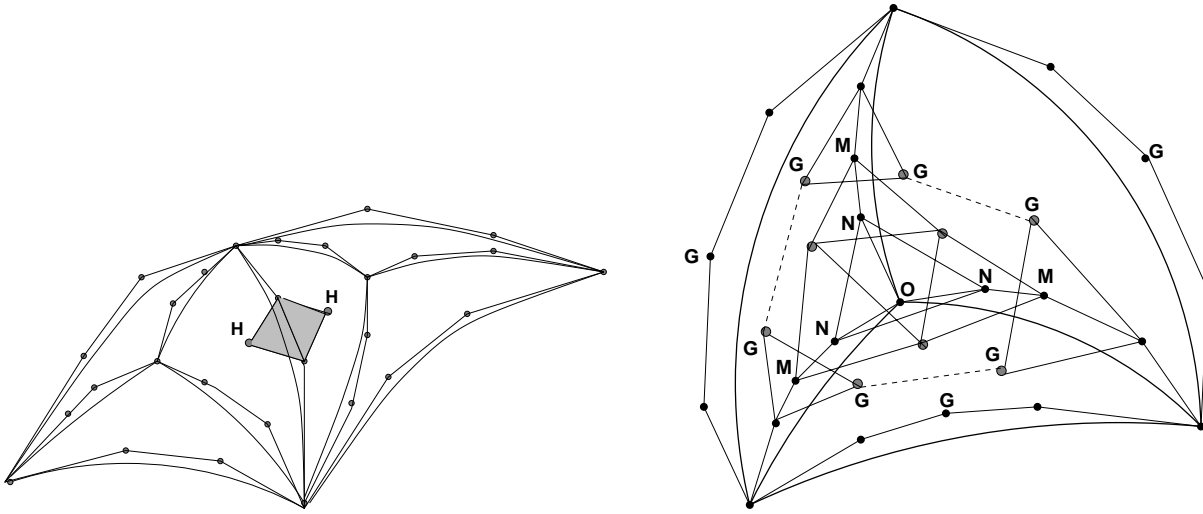


FIG. 2.4 – *A gauche : une correction est portée sur les points répertoriés par \mathbf{H} , afin de rendre les deux triangles hachurés coplanaires. A droite : les points \mathbf{G} sont modifiés de manière à obtenir une continuité G^1 et approcher au mieux les points “candidats” et les point $\mathbf{M}, \mathbf{N}, \mathbf{O}$ doivent être modifiés aux centroïdes des triangles qui les entourent afin d’assurer une continuité C^1 le long des courbes internes.*

L’idée de base de cette approche va dans la suite être exposée par la méthode de Gregory [56]. La méthode “side-vertex” de Nielson [102, 104] est un autre exemple de ce type de méthodes. Elle est, cependant, très connue dans un autre domaine voisin : l’interpolation des données dispersées (scattered data interpolation) non-paramétriques avec le “Minimum Norm Network” méthode [103]. Les travaux de Gregory [56, 59, 57, 58, 60] dont nous présentons ici les principes de bases sont connus pour avoir étudié le twist discontinu pour ce type d’interpolants.

Pour d’autres méthodes de ce genre, voir Farin [43], Gregory et Charrot [59], Barnhill [5] ou Mann [89].

2.6.1 Méthode de Gregory

La méthode de Gregory construit d’abord un réseau de courbes frontière cubiques correspondant aux arêtes de \mathcal{M} , puis elle applique la méthode de Chiyokura-Kimura (sect. 1.3.3) le long des courbes frontière de chaque facette M afin d’engendrer les dérivées trans-frontière assurant une continuité G^1 . On obtient alors les six points de contrôle intérieurs \mathbf{p}_{ij} , $i, j = 1, 2, 3$, $i \neq j$, montrés fig. 2.5. Ces points s’appellent les points “twist” de M .

Nous avons vu, sect. 2.4 que la représentation polynomiale de la facette présente le

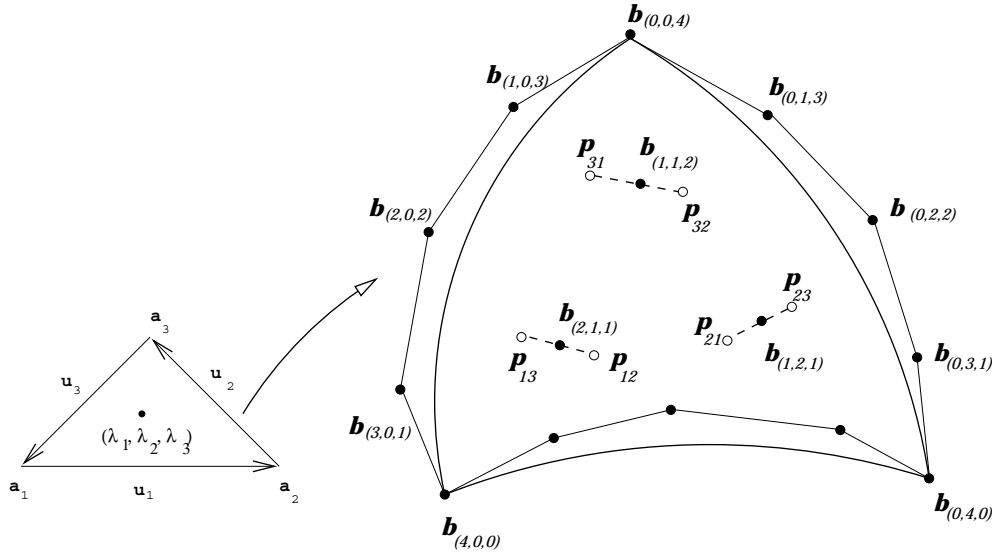


FIG. 2.5 – Facette de Gregory triangulaire.

problème de twist. Pour résoudre ce problème, Gregory propose de construire une facette quartique rationnelle M de twist discontinu, en introduisant des termes rationnels, i.e. :

$$M(\lambda_1, \lambda_2, \lambda_3) = \sum_{p+q+r=4} B_{(p,q,r)}^4(\lambda_1, \lambda_2, \lambda_3) \mathbf{b}_{(p,q,r)}^4, \quad (2.3)$$

où $(\lambda_1, \lambda_2, \lambda_3)$ sont les coordonnées barycentriques d'un point quelconque de Δ et $\mathbf{b}_{(p,q,r)}$ sont les points de contrôle de M . Ils sont tous constants sauf les trois points $\mathbf{b}_{(1,1,2)}$, $\mathbf{b}_{(2,1,1)}$ et $\mathbf{b}_{(1,2,1)}$. Chacun de ces points est une combinaison convexe des 2 points "twist" obtenus par la méthode de Chiyokura-Kimura (liés par une ligne coupée dans la figure 2.5), i.e. :

$$\begin{cases} \mathbf{b}_{(1,1,2)} = \frac{\lambda_1(1-\lambda_2)\mathbf{p}_{31} + \lambda_2(1-\lambda_1)\mathbf{p}_{32}}{\lambda_1(1-\lambda_2) + \lambda_2(1-\lambda_1)}, \\ \mathbf{b}_{(2,1,1)} = \frac{\lambda_3(1-\lambda_2)\mathbf{p}_{13} + \lambda_2(1-\lambda_3)\mathbf{p}_{12}}{\lambda_3(1-\lambda_2) + \lambda_2(1-\lambda_3)}, \\ \mathbf{b}_{(1,2,1)} = \frac{\lambda_1(1-\lambda_3)\mathbf{p}_{21} + \lambda_3(1-\lambda_1)\mathbf{p}_{23}}{\lambda_1(1-\lambda_3) + \lambda_3(1-\lambda_1)}. \end{cases} \quad (2.4)$$

Le calcul du twist de M au sommet \mathbf{a}_1 du triangle domaine donne :

$$\begin{cases} [D_{\mathbf{u}_1} D_{-\mathbf{u}_3}] M(0,0) = 12(\mathbf{b}_{(3,0,1)} + \mathbf{p}_{12} - \mathbf{b}_{(4,0,0)} - \mathbf{b}_{(3,1,0)}), \\ [D_{-\mathbf{u}_3} D_{\mathbf{u}_1}] M(0,0) = 12(\mathbf{b}_{(3,0,1)} + \mathbf{p}_{13} - \mathbf{b}_{(4,0,0)} - \mathbf{b}_{(3,1,0)}). \end{cases} \quad (2.5)$$

La facette M a été conçue pour avoir les 3 vecteurs twists discontinus. Ce type de facettes n'a donc pas de problème avec le twist. La G^1 continuité de la surface finale résulte de l'utilisation de la méthode de Chiyokura-Kimura pour engendrer les dérivées trans-frontière,

assurant une continuité de plan tangent entre les facettes voisines. Notons aussi que le calcul des dérivées trans-frontière le long d'une courbe se fait d'une manière indépendante des courbes adjacentes, ce qui reflète la stricte localité de la méthode.

Cette méthode de Gregory construit une surface interpolante G^1 continue d'une manière simple et élégante. Mais du côté pratique, il faut noter qu'une facette de Gregory peut être vue comme une combinaison convexe de sept facettes quartiques de Bézier et les facteurs de cette combinaison convexe sont des fonctions rationnelles avec des polynômes de degré six (voir Mann et al. [92], page 157).

2.7 Approche de facettes dégénérées

Une autre manière de résoudre le problème d'interpolation avec des surfaces polynomiales est l'utilisation des *facettes triangulaires de Bézier dégénérées*. Une facette triangulaire de Bézier dégénérée est une facette triangulaire de Bézier dont les trois courbes frontière sont dégénérées d'un degré aux trois sommets. La dégénérescence d'un degré d'une courbe signifie qu'elle confond les deux premiers et les deux derniers de ces points de Bézier. Nous avons étudié la méthode de Neamtu et Pfluger [98] mais l'idée de l'utilisation des facettes dégénérées a été proposée avant par Peters [109] et Du et Schmitt [30]. Cette méthode construit une seule facette quintique dégénérée pour chaque face. L'hypothèse de la dégénérescence est suffisante pour avoir une solution au problème de twist. En plus, les vecteurs normaux peuvent être interpolés avec les sommets tout en gardant un schéma strictement local. Une minimisation d'une fonction quadratique est utilisée pour définir les 6 points intérieurs de chaque patch.

Enfin, il est à noter que ces facettes possèdent les mêmes propriétés géométriques que les facettes de Bézier (invariance affine, invariance par transformation affine de paramètres, enveloppe convexe, etc).

2.8 Approche des facettes algébriques

Une facette algébrique M est définie sur Δ par une équation implicite $M(x, y, z) = 0$, où $M(x, y, z)$ est un polynôme en (x, y, z) . Le degré total de ce polynôme est le degré de la facette implicite. Si n est le degré de M alors $M(x, y, z)$ a au maximum C_{n+3}^3 coefficients.

Toutes les méthodes que nous avons vues jusqu'ici construisent des surfaces paramétriques. Il y a peu de méthodes qui utilisent les surfaces polynomiales implicites pour résoudre le problème d'interpolation. Sederberg [122] a montré plusieurs surfaces implicites lisses sous forme de Bernstein à trois variables et qui peuvent être manipulées comme des fonctions

de Bézier par un réseau de contrôle constitué de tétraèdres. Nous avons étudié la méthode de Bajaj et Ihm [4] qui construit pour chaque face une seule facette polynomiale implicite de degré 7. Elle offre au moins cinq degrés de liberté pour contrôler la forme de chaque facette. Une approximation au sens des moindres carrés est ensuite utilisée pour fixer ces libertés dans une procédure d'optimisation de forme. Nous ne détaillerons pas cette méthode mais notons seulement que la surface produite par cette méthode présente des singularités aux sommets et c'est de cette manière que le problème de twist est résolu.

2.9 Approche des courbes frontière

Toutes les méthodes que nous avons exposées construisent d'abord un réseau de courbes correspondant aux arêtes de \mathcal{M} . La construction de courbes frontière C^2 -compatibles est une technique récente pour la solution du problème de twist. Cette technique consiste à construire les courbes frontière dans une variété de dimension 2 de continuité C^2 . Dans les sections suivantes, nous allons exposer deux méthodes type suivant cette approche : la méthode de Loop dans la section 2.9.1 et la méthode de Peters dans la section 2.9.2.

2.9.1 Méthode de Loop

Loop [84] a proposé une méthode qui, à chaque face de \mathcal{M} , fait correspondre une seule facette triangulaire de Bézier de degré six. Cette méthode a été conçue pour l'approximation et non pas pour l'interpolation des sommets du réseau surfacique. L'interpolation est théoriquement possible mais mène à des ondulations des courbes frontière. La méthode procède en trois étapes :

- construction de courbes frontière,
- construction des tangentes trans-frontière,
- remplissage des facettes.

Dans la première étape, la méthode construit le réseau des courbes frontière de manière à résoudre le problème de twist.

Dans la suite de cette section, nous exposerons brièvement la méthode de Loop et nous montrerons qu'elle ne convient pas pour l'interpolation.

Notations

Les facettes et les courbes frontière sont respectivement en correspondance bijective avec les triangles et les arêtes de \mathcal{M} . Pour faciliter la présentation, \mathcal{M} est supposé être fermé. Il est donc naturel de considérer la construction des facettes autour de chaque sommet de \mathcal{M} . Ces facettes partageant deux à deux une frontière commune et chacune

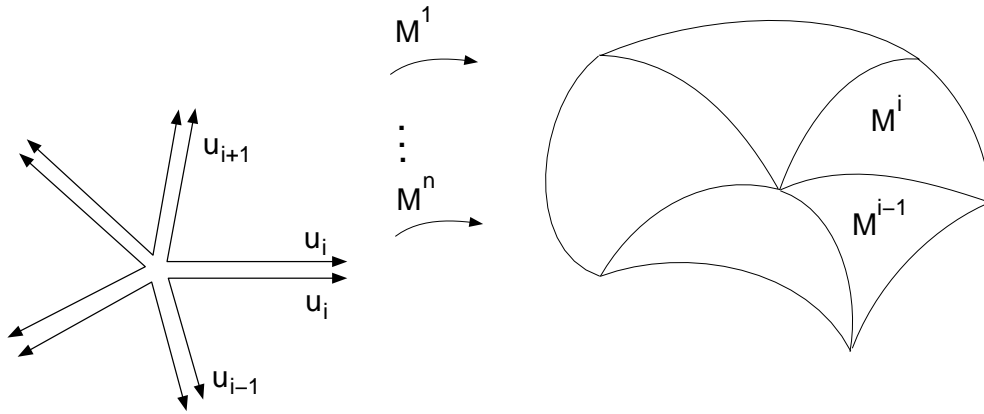


FIG. 2.6 – Paramétrisation des n facettes adjacentes au sommet \mathbf{v} .

est l'image polynomiale du triangle Δ . Considérons donc un sommet quelconque \mathbf{v} de degré n et les facettes qui l'entourent M^i , $i = 1, \dots, n$. La figure 2.6 montre la notation utilisée pour la paramétrisation de ces facettes.

Les conditions nécessaires et suffisantes pour un raccordement G^1 entre deux facettes M^{i-1} et M^i le long de leur courbe commune sont (voir 1.3.3)

- $M^{i-1}(0, u_i) = M^i(u_i, 0)$, $\forall u_i \in [0, 1]$,
- existence de trois fonctions réelles Φ_i , μ_i et ν_i telles que

$$\begin{aligned} \Phi_i(u_i) [D_{\mathbf{u}_i} M^i](u_i, 0) &= \mu_i(u_i) [D_{\mathbf{u}_{i-1}} M^{i-1}](0, u_i) \\ &\quad + \nu_i(u_i) [D_{\mathbf{u}_{i+1}} M^i](u_i, 0), \end{aligned} \tag{2.6}$$

$$[D_{\mathbf{u}_i} M^i](u_i, 0) \times [D_{\mathbf{u}_{i+1}} M^i](u_i, 0) \neq 0,$$

$$\mu_i(u_i)\nu_i(u_i) > 0, \quad \forall u_i \in [0, 1].$$

Le problème de twist (voir 2.4) apparaît ici lorsqu'on dérive la première équation de 2.6 dans la direction de \mathbf{u}_i et puis l'évalue en $u_i = 0$, on obtient le système suivant à n équations linéaires et n vecteurs twists inconnus :

$$\begin{aligned} \mu_i(0)D_{\mathbf{u}_{i-1}}D_{\mathbf{u}_i}M^{i-1}(0, 0) + \nu_i(0)D_{\mathbf{u}_i}D_{\mathbf{u}_{i+1}}M^i(0, 0) &= \\ \Phi'_i(0)D_{\mathbf{u}_i}M^i(0, 0) + \Phi_i(0)D_{\mathbf{u}_i}D_{\mathbf{u}_i}M^i(0, 0) + & \\ \nu'_i(0)D_{\mathbf{u}_{i+1}}M^i(0, 0) + \mu'_i(0)D_{\mathbf{u}_{i-1}}M^{i-1}(0, 0), & \end{aligned} \tag{2.7}$$

$$i = 1, \dots, n.$$

On remarque que le deuxième membre de ce système ne dépend que des dérivées premières et secondes des courbes frontière autour de \mathbf{v} . Donc si les courbes frontière sont

fixées, le deuxième membre de 2.7 l'est aussi, et on peut résoudre 2.7 pour les n vecteurs twist autour de \mathbf{v} . Mais ce système linéaire possède une matrice circulante singulière⁴ si n est pair (Van Wijk [129], Peters[110]). Ceci reflète le problème de compatibilité de twist. Loop propose donc une construction particulière des courbes frontière, telle que le membre droit de 2.7 soit dans l'espace image de la matrice. Ainsi le problème de twist est résolu car il existe toujours une solution au système.

La stratégie de Loop consiste donc à construire les courbes frontière autour de \mathbf{v} pour que le système (2.7) possède toujours une solution. Elle peut se résumer aux étapes suivantes :

Choix des fonctions réelles

Loop prend les fonctions réelles suivantes :

$$\begin{cases} \Phi_i(t) = \cos \frac{2\pi}{n} B_0^2(t) + \frac{1}{2} B_1^2(t) + (1 - \cos \frac{2\pi}{n}) B_2^2(t) & (\text{quadratique}) \\ \mu_i(t) = \nu_i(t) = \frac{1}{2} & (\text{constantes}) \end{cases} \quad (2.8)$$

Pour $t = 0$, on remarque que toutes les fonctions Φ_i prennent une même valeur $\Phi_i(0) = \cos \frac{2\pi}{n}$, indépendamment de i . De même, $\Phi'_i(0) = 1 - 2 \cos \frac{2\pi}{n}$ ne dépend pas de i . Ceci va simplifier les conditions du raccordement G^1 .

Notations :

Posons

$$\Phi^0 = \Phi_i(0), \quad \Phi^1 = \Phi'_i(0),$$

$$\mathbf{r}_i^1 = [D_{\mathbf{u}_i}] M^i(0, 0),$$

$$\mathbf{r}_i^2 = [D_{\mathbf{u}_i} D_{\mathbf{u}_i}] M^i(0, 0),$$

$$\mathbf{t}_i = [D_{\mathbf{u}_i} D_{\mathbf{u}_{i+1}}] M^i(0, 0), \text{ pour } i = 1, \dots, n.$$

Les conditions entre les dérivées premières données par la première équation de 2.6 s'écrivent matriciellement sous la forme suivante :

$$\begin{bmatrix} \Phi^0 & -\frac{1}{2} & \dots & & -\frac{1}{2} \\ & -\frac{1}{2} & \Phi^0 & -\frac{1}{2} & \\ & & \ddots & \ddots & \ddots \\ & & & & -\frac{1}{2} & \Phi^0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^1 \\ \mathbf{r}_2^1 \\ \vdots \\ \mathbf{r}_n^1 \end{bmatrix} = 0 \quad (2.9)$$

4. elle est de rang égal à $n - 1$ si n est pair.

et les contraintes de twists données par 2.7 s'écrivent

$$\begin{bmatrix} \frac{1}{2} & 0 & \dots & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \dots & 0 \\ & \ddots & \ddots & \\ 0 & \dots & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \dots & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_{n-1} \\ \mathbf{t}_n \end{bmatrix} = \Phi^1 \begin{bmatrix} \mathbf{r}_1^1 \\ \mathbf{r}_2^1 \\ \vdots \\ \mathbf{r}_{n-1}^1 \\ \mathbf{r}_n^1 \end{bmatrix} + \Phi^0 \begin{bmatrix} \mathbf{r}_1^2 \\ \mathbf{r}_2^2 \\ \vdots \\ \mathbf{r}_{n-1}^2 \\ \mathbf{r}_n^2 \end{bmatrix}, \quad (2.10)$$

Le choix précédent de Φ_i (donné par 2.8) est motivé par les trois remarques suivantes :

- Le déterminant du système homogène 2.9 doit s'annuler pour obtenir une solution non triviale des vecteurs tangents. Ce déterminant est égal à $(-1)^n \prod_{k=0}^{n-1} (\cos(\frac{2\pi k}{n}) - \Phi^0)$. Il s'annule quand $\Phi^0 = \Phi_i(0) = \cos(\frac{2\pi}{n})$.
- Afin de calculer la valeur de Φ_i en $t = 1$, on applique le même raisonnement sur le sommet \mathbf{p}_i , directement voisin à \mathbf{v} , appelé aussi sommet opposé (voir fig. 2.7). On trouve alors $\Phi_i(1) = 1 - \cos(\frac{2\pi}{n_i})$, $i = 1, \dots, n$, où n_i est le degré du sommet opposé. Pour plus de détail, voir proposition 2 du chapitre suivant.
- Si $n = n_i$, alors $\Phi_i(t) = \Phi_i^0(1-t) + (1 - \Phi_i^0)t$ et Φ_i devient une fonction linéaire. Et si de plus $n = n_i = 6$ alors $\Phi_i \equiv \frac{1}{2}$.

Le degré de quelques facettes peut ainsi être réduit lorsque leurs sommets présentent une certaine régularité, voir Loop [84].

Construction des courbes frontière

Les conditions requises sur les courbes frontière sont les suivantes :

- interpolation des sommets de \mathcal{M} ,
- réalisation des conditions G^1 ,
- garder l'aspect local de l'algorithme.

Appelons \mathbf{p}_i , $i = 1, \dots, n$, les sommets voisins de \mathbf{v} et n_i leurs degrés respectifs. Notons la courbe frontière allant de \mathbf{v} à \mathbf{p}_i par sa forme de Bézier quartique $\mathbf{b}_0^i, \dots, \mathbf{b}_4^i$, voir fig. 2.7. Autour de chaque sommet de \mathcal{M} , les points de contrôle $\mathbf{b}_0^i, \mathbf{b}_1^i, \mathbf{b}_2^i$, $i = 1, \dots, n$ sont construits de manière à ce que les dérivées premières et secondes des courbes frontière se trouvent dans l'espace image de la matrice du système 2.7, Loop trouve la solution suivante :

$$\begin{aligned} \mathbf{b}_0^i &= \alpha \mathbf{v} + \frac{(1-\alpha)}{n} \sum_{j=1}^n \mathbf{p}_j \\ \mathbf{b}_1^i &= \alpha \mathbf{v} + \frac{1}{n} \sum_{j=1}^n (1 - \alpha + \beta \cos(\frac{2\pi(j-i)}{n})) \mathbf{p}_j \\ \mathbf{b}_2^i &= \frac{1}{3} \mathbf{v} + \frac{1}{6} \mathbf{p}_{i-1} + \frac{1}{3} \mathbf{p}_i + \frac{1}{6} \mathbf{p}_{i+1}. \end{aligned} \quad (2.11)$$

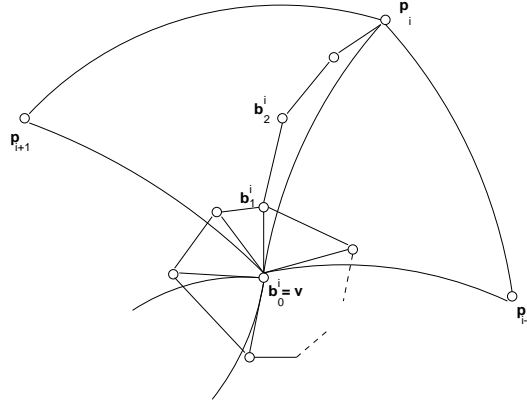


FIG. 2.7 – points de contrôle des courbes frontière incidentes en un sommet \mathbf{v} .

Les deux points de contrôle $\mathbf{b}_3^i, \mathbf{b}_4^i$ sont construits de manière analogue à partir du sommet opposé \mathbf{p}_i . α et β sont deux paramètres de forme : le paramètre α exprime la manière d'approcher ou d'interpoler les sommets du réseau. Le choix $\alpha = 1$ implique l'interpolation. β est un paramètre de forme qui contrôle les longueurs des vecteurs tangents. On remarque que dans le cas de l'interpolation, nous n'avons pas de liberté dans la construction de \mathbf{b}_2^i , car il est construit d'une manière symétrique des deux sommets \mathbf{v} et \mathbf{p}_i . C'est pour cela que les courbes frontière peuvent effectuer des ondulations indésirables.

Construction des tangentes trans-frontière

Les tangentes trans-frontière le long des courbes frontière assurant la continuité G^1 entre les patches sont construites en utilisant des conditions G^1 équivalentes⁵:

$$\begin{aligned} D_{\mathbf{u}_{i+1}} M^i(u_i, 0) &= \Phi_i(u_i) D_{\mathbf{u}_i} M^i(u_i, 0) + \Psi_i(u_i) \mathbf{V}_i(u_i), \\ D_{\mathbf{u}_{i-1}} M^{i-1}(0, u_i) &= \Phi_i(u_i) D_{\mathbf{u}_i} M^i(u_i, 0) - \Psi_i(u_i) \mathbf{V}_i(u_i). \end{aligned} \quad \forall u_i \in [0, 1], \quad (2.12)$$

La fonction réelle Ψ_i et la fonction vectorielle \mathbf{V}_i sont construites de bas degré afin d'interpoler les valeurs des vecteurs tangents et des twists aux sommets \mathbf{v} et \mathbf{p}_i :

$$\begin{aligned} \Psi_i(u_i) &= \sin\left(\frac{2\pi}{n}\right)(1 - u_i) + \sin\left(\frac{2\pi}{n_i}\right)u_i \quad (\text{linéaire}) \\ \mathbf{V}(u_i) &= \sum_{k=1}^3 \mathbf{v}_k^i B_3^k(u_i) \quad (\text{cubique}). \end{aligned} \quad (2.13)$$

Les deux points de contrôle \mathbf{v}_2^i et \mathbf{v}_3^i de \mathbf{V}_i sont donnés avec les mêmes formules appliquées au sommet opposé \mathbf{p}_i .

Le degré des tangentes trans-frontière est égal à 5. Donc les facettes M^i sont de degré 6. On doit donc élever le degré des courbes frontière de 4 à 6.

5. voir les conditions 1.16, sect. 1.3.2 avec $p_1 = \Psi_i$, $p_2 = -\Psi$ et $q_1 = q_2 = \Phi_i$.

Remplissage des facettes

Chaque facette est donnée par ses points de contrôle. La rangée frontière et la première rangée intérieure des points de contrôle sont obtenues à partir des courbes frontière et des tangentes trans-frontière. Ces points assurent un raccordement G^1 entre les facettes voisines. Reste à déterminer un seul point de contrôle intérieur pour chaque facette. Ce dernier est construit tel que si tous les autres points de contrôle proviennent d'une facette quintique, élevée en degré, alors la facette sera quintique. On dit que la facette est de précision quintique. Dans [84], page 323, Loop donne la formule exacte pour la détermination de ce point.

Notons que le degré polynomial d'une facette est réduit à 5 si les trois sommets de cette facette ont le même degré. Et il devient 4 s'ils sont de degré 6.

Interpolation

L'interpolation est théoriquement possible, avec $\alpha = 1$, mais en pratique mène à des ondulations indésirables de la surface, car la méthode impose des contraintes sévères sur les dérivées secondes aux extrémités des courbes frontière. La figure 2.8 montre l'exemple de l'interpolation des sommets d'un icosaèdre avec la méthode de Loop. En bas de la figure, on voit un polygone de contrôle et l'arête correspondante extraits du réseau. Il s'avère que le polygone n'est pas convexe dû au choix du point \mathbf{b}_2 bien que le réseau le soit parfaitement. Ainsi les courbes frontière peuvent présenter des points d'inflexion indésirables. C'est pourquoi le réseau surfacique de Loop se présente comme un réseau approché et non pas interpolé.

2.9.2 Méthode de Peters

La méthode de Peters [110] construit une facette polynomiale régulière pour chaque face du réseau surfacique. Elle part de la constatation qu'il n'y a un problème de twist qu'aux sommets de degré pair et que ce problème est posé quel que soit le nombre d'arêtes des facettes autour de ce sommet. Peters propose donc des solutions pour ces sommets de degré pair suivant leur configuration. Ici nous exposons seulement le problème avec des facettes triangulaires et nous nous intéressons plus particulièrement à la manière de résolution du problème de twist et le degré des facettes utilisées. Peters impose la contrainte suivante sur les courbes frontière :

Contrainte C (Contrainte de courbure) : les courbes autour de \mathbf{v} sont compatibles avec une 2^{ème} forme fondamentale donnée en \mathbf{v} par deux courbures principales et une direction principale.

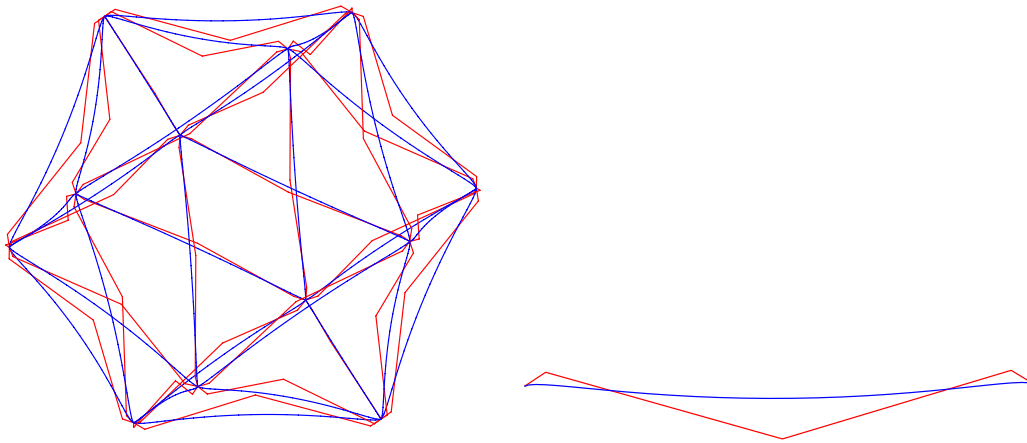


FIG. 2.8 – En haut : construction des courbes frontière avec la méthode de Loop appliquée pour l'interpolation des sommets d'un icosaèdre. En bas : une courbe extraite du réseau de courbes précédent.

Cette contrainte oblige les courbes aux frontières partant de \mathbf{v} d'être compatibles avec deux courbures principales et une direction principale spécifiées en chaque sommet du réseau. Avec cette contrainte, Peters arrive à résoudre le problème de twist en ajustant les fonctions réelles dans les conditions du raccordement G^1 .

Problème de twist

Nous reprenons les notations de la section précédente (2.9.1). Le raccordement G^1 entre les deux facettes M^{i-1} et M^i le long de leur courbe commune est assurée par les conditions 2.6. En particulier l'évaluation de la première condition en $u_i = 0$ donne :

$$\Phi^0 \mathbf{r}_i^1 = \mu_i^0 \mathbf{r}_{i-1}^1 + \nu_i^0 \mathbf{r}_{i+1}^1. \quad (2.14)$$

Les contraintes de twist sont données par :

$$\mu_i^0 \mathbf{t}_{i-1} + \nu_i^0 \mathbf{t}_i = \Phi_i^1 \mathbf{r}_i^1 + \Phi_i^0 \mathbf{r}_i^2 + \nu_i^1 \mathbf{r}_{i+1}^1 + \mu_i^1 \mathbf{r}_{i-1}^1, \quad i = 1, \dots, n. \quad (2.15)$$

Si on pose $\Phi_i^0 := \rho_i \eta_i$, $\mu_i^0 = \rho_i k_i$ et $\nu_i^0 = \rho_i (1 - k_i)$, avec $\rho_i > 0$, alors η_i et k_i se déterminent de $\eta_i \mathbf{r}_i^1 = k_i \mathbf{r}_{i-1}^1 + (1 - k_i) \mathbf{r}_{i+1}^1$ et l'équation 2.15 s'écrit :

$$k_i \mathbf{t}_{i-1} + (1 - k_i) \mathbf{t}_i = \mathbf{m}_i, \quad (2.16)$$

où $\mathbf{m}_i = (1/\rho_i)(\Phi_i^1 \mathbf{r}_i^1 + \Phi_i^0 \mathbf{r}_i^2 + \nu_i^1 \mathbf{r}_{i+1}^1 + \mu_i^1 \mathbf{r}_{i-1}^1)$. Pour résoudre le problème de twist, Peters propose une stratégie différente de celle de Loop. Il construit d'abord un réseau de courbes

frontière C^2 -compatibles aux sommets, ensuite il construit les fonctions réelles Φ_i , μ_i et ν_i de manière à ce que le deuxième membre du système 2.7 est dans l'espace image de la matrice des coefficients.

En effet, soit \mathbf{v} un sommet quelconque de degré n , et \mathbf{n} son vecteur normal. Sachant que les courbes frontière autour de \mathbf{v} sont compatibles avec une certaine deuxième forme fondamentale, alors il existe une matrice symétrique Π d'ordre 2, telle que

$$\mathbf{n} \cdot \mathbf{r}_i^2 = -\mathbf{r}_i^1 \Pi \mathbf{r}_i^1 \quad i = 1, \dots, n.$$

En faisant le produit scalaire des deux membres du système 2.15 avec \mathbf{n} , on obtient

$$-\Phi_i^0 \mathbf{r}_i^1 \Pi \mathbf{r}_i^1 = \mu_i^0 \mathbf{n} \cdot \mathbf{t}_{i-1} + \nu_i^0 \mathbf{n} \cdot \mathbf{t}_i, \quad i = 1, \dots, n. \quad (2.17)$$

Pour résoudre le problème de twist, on commence par choisir la composante normale de \mathbf{t}_i de la manière suivante

$$\mathbf{n} \mathbf{t}_i = -\mathbf{r}_i^1 \Pi \mathbf{r}_{i+1}^1.$$

De cette manière, les conditions normales du twist s'écrivent :

$$[\Phi_i^0 \mathbf{r}_i^1 - \mu_i^0 \mathbf{r}_{i-1}^1 - \nu_i^0 \mathbf{r}_{i+1}^1] \Pi \mathbf{r}_i^1 = 0. \quad (2.18)$$

Mais ces conditions sont automatiquement vérifiées à partir de 2.14.

Reste à satisfaire les deux composantes tangentiels du système 2.15. Pour les vérifier, Peters montre que les coefficients Φ_i^1 peuvent être toujours déterminés afin que les composantes tangentiels de $\bar{\mathbf{m}} = (\mathbf{m}_1, \dots, \mathbf{m}_n)^T$ appartiennent à l'espace image de la matrice des coefficients du système 2.16. Par conséquent, la fonction Φ_i doit être construite cubique, car la valeur de sa dérivée en $u_i = 0$ est utilisée pour satisfaire les contraintes de twist. Cette valeur doit être indépendante de celle en $u_i = 1$, sinon le problème de twist au sommet opposé \mathbf{p}_i sera mélangé avec celui en \mathbf{v} .

Peters montre aussi que les facettes quartiques ne sont pas suffisantes pour résoudre le problème du raccordement G^1 et qu'il faut utiliser des facettes quintiques, dans le sens suivant : si les points de contrôle des deux premières rangées de la facette appartiennent à une facette quintique alors la présente facette sera aussi quintique.

Reste à répondre à la question : comment construire des courbes frontière C^2 -compatibles ? Les travaux de De Boor, Höllig et Sabin ([27], [75]) ont montré que les courbes cubiques sont incapables de produire toujours une interpolation de position, de tangente et de courbure. Peters utilise les courbes quartiques pour l'interpolation d'une deuxième forme fondamentale en chaque sommet du réseau surfacique.

Construction de courbes frontière C^2 -compatibles

Soient κ_1, κ_2 les deux courbures principales spécifiée au sommet \mathbf{v} et $\mathbf{E}_1, \mathbf{E}_2$ les deux directions principales correspondantes. La courbure normale de la courbe frontière $M^i(u_i, 0)$ est donnée par la formule d'Euler

$$\kappa^i = \kappa_1(\mathbf{E}_1 \cdot \mathbf{r}_i^1)^2 + \kappa_2(\mathbf{E}_2 \cdot \mathbf{r}_i^1)^2.$$

La courbe frontière $M^i(u_i, 0)$ est construite sous forme de Bézier quartique: $M^i(u_i, 0) = \sum_{k=0}^4 \mathbf{b}_k^i B_k^4(u_i)$ avec les contraintes suivantes :

- construire d'abord une courbe cubique avec une méthode similaire à celle de Piper (voir sect. 2.5.1), ensuite faire une élévation de degré de cette courbe et fixer le point de contrôle \mathbf{b}_2^i par le point de contrôle correspondant de la courbe ainsi obtenue, voir fig. 2.9.

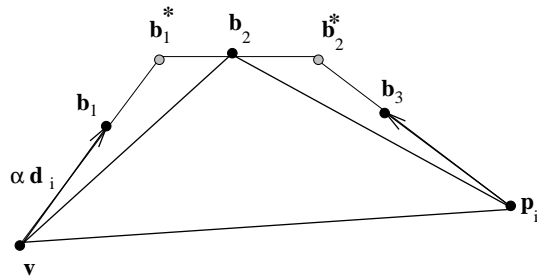


FIG. 2.9 – Construction de courbes C^2 -compatibles.

- choisir le point de contrôle $\mathbf{b}_1^i = \mathbf{v} + \alpha \mathbf{d}_i$, avec \mathbf{d}_i la tangente en \mathbf{v} estimée par la méthode de Piper et α est un paramètre réel déterminé d'une manière à interpoler la courbure normale κ_i en \mathbf{v} dans la direction de \mathbf{d}_i . Donc α est calculé de l'équation

$$\kappa_i = \frac{\|\mathbf{r}_i^1 \times \mathbf{r}_i^2\|}{\|\mathbf{r}_i^1\|^3} = \frac{\|4\alpha \mathbf{d}_i \times 12(\mathbf{b}_2^i - \mathbf{b}_1^i - \alpha \mathbf{d}_i)\|}{4^3 \alpha^3 \|\mathbf{d}_i\|^3} = \frac{3\|\mathbf{d}_i \times (\mathbf{b}_2^i - \mathbf{v})\|}{4\alpha^2}.$$

D'où

$$\mathbf{b}_1^i = \mathbf{v} + \sqrt{\frac{3\|\mathbf{d}_i \times (\mathbf{b}_2^i - \mathbf{v})\|}{4\kappa_i}} \mathbf{d}_i.$$

Si $\kappa_i = 0$, α est fixé arbitrairement et $\mathbf{d}_i = (\mathbf{b}_2^i - \mathbf{v}) / \|\mathbf{b}_2^i - \mathbf{v}\|$, i.e. dans ce cas, les trois points $\mathbf{b}_0^i, \mathbf{b}_1^i$ et \mathbf{b}_2^i sont alignés.

- le point de contrôle \mathbf{b}_3^i est construit de la même manière à partir du sommet opposé.
- Dès qu'un réseau de courbes C^2 -compatibles est construit, la méthode calcule les fonctions Φ_i, μ_i et ν_i sous les contraintes 2.6 et 2.14, ensuite elle calcule les vecteurs twist compatibles

à partir des équations 2.15. Les dérivées trans-frontière sont ensuite calculées sous forme de Bézier quartique et les facettes sont quintiques.

Remarque 5 *La technique de subdivision de type Clough-Tocher peut fonctionner avec cette méthode, et permet de baisser le degré des facettes à 3, voir Peters [106].*

2.10 Discussion

Le long des sections 2.5 - 2.9, nous avons exposé les idées de base des différentes méthodes d'interpolation paramétrique. Elles se différencient dans la façon dont elles produisent une continuité G^1 globale de la surface finale. Le problème de twist ne se pose pas pour les méthodes se basant sur des facettes rationnelles ou dégénérées aux sommets. Mais pour les surfaces polynomiales non dégénérées le problème de twist est résolu soit par subdivision, par des courbes de bord compatibles avec une certaine deuxième forme fondamentale ou par des courbes de bord particulières à la manière de Loop. Un autre aspect qui n'a pas encore été mentionné est la qualité des surfaces produites par ces méthodes. En effet, la forme de la surface finale est important pour les utilisateurs et après avoir construit la surface finale, l'utilisateur peut donner un jugement sur la forme finale de la surface, comme par exemple "forme agréable" ou "forme esthétique". D'où l'intérêt d'un grand nombre de recherches pour les critères de lissage ("fairness") et leurs caractérisations mathématiques, en développant un ensemble de méthodes permettant de dire si une surface est lisse ou non suivant un certain critère fonctionnelle. Ces critères englobent un grand nombre de caractéristiques géométriques de la surface, comme par exemple : la convexité, la diminution des variations de la courbure, la réduction des bosses, des oscillations, des coins, ... Le principe de la forme la plus simple est très souvent utilisé pour établir ces critères, signifiant qu'une surface est lisse si elle ne contient que des formes simples. Un critère de lissage peut être appliqué à une surface produite par une méthode (d'interpolation) capable d'offrir des paramètres libres, pouvant être utilisés soit localement ou globalement pour satisfaire la forme désirée de la surface. Pour des détails sur les critères de lissage voir Roulier, Rando [116], Burchard et al. [21], Moreton, Séquin [97] et les références qui y sont citées.

A cause de notre hypothèse de réseau surfacique de topologie arbitraire, qui est une contrainte sévère, les méthodes présentées dans ce chapitre souffrent toutes de certains défauts qui peuvent être vus sur la surface interpolante. Le résumé de Mann et al. [92] contient une discussion des inconvénients de la plupart des méthodes citées ici. Il en résulte que

- de manière générale, le problème d'interpolation avec continuité G^1 reste un problème difficile et contraint l'obtention de résultats satisfaisants en terme de la qualité définie

ci-dessus.

- En particulier, les méthodes de subdivision souffrent du fait que les angles des triangles du réseau surfacique sont divisés par deux, ce qui allonge les sous facettes deux fois plus que les faces de départ. Cela peut mener à des triangles très allongés pouvant créer des oscillations indésirables sur la surface finale. Par contre, ces méthodes sont les seules à permettre de choisir les courbes frontière sans contraintes et permettent aussi l'utilisation des facettes polynomiales de plus bas degré.
- les méthodes de combinaison convexes ont l'inconvénient de donner des surfaces rationnelles de degré assez élevé. Il apparaît d'après Mann et al. [92] que les patches de Gregory permettent d'obtenir des qualités à peu près satisfaisantes. Dans Du [31], on peut même voir des exemples. Mais tout calcul sur la surface (dérivation, courbures, lignes de réflexion, etc) devient très lourd. D'après ce qu'on sait, ces surfaces ne se sont jamais imposées dans la pratique.
- Les facettes algébriques ou dégénérées sont très originales du point de vue théorique. Les premières ont une définition implicite et non explicite et nécessitent un environnement de calcul spécifique. Les deuxièmes ont l'inconvénient d'imposer une contrainte sévère sur les twists ce qui peut rendre difficile la modélisation de surfaces de topologies quelconques. Par contre, les résultats sur des réseaux de configuration convexe semblent être satisfaisants.
- La méthode de Loop est une approche à part, car elle n'a pas été conçue au départ pour l'interpolation mais pour l'approximation du réseau surfacique. Donc si on impose l'interpolation, les courbes frontière peuvent présenter des ondulations réduisant ainsi la qualité générale de la surface. Pour l'approximation, elle donne de bons résultats et la résolution du problème de twist est originale. Cette méthode a initialisé le développement de la plus récente méthode du 4-split (voir chap. 3). Elle est donc en relation très étroite avec les contributions qui vont être présentées dans cette thèse.
- La méthode de Peters impose sur les courbes frontière la contrainte d'être C^2 -compatibles avec les deuxièmes formes fondamentales spécifiées aux sommets. Sachant la sensibilité des calculs vis à vis de la courbure surfacique, cette méthode ne possède pas suffisamment de paramètres libres afin de contrôler localement la forme de la surface finale. Sinon, la méthode présente une élégance théorique et une originalité dans la résolution du problème de twist, avec des facettes triangulaires et/ou quadrilatères.

2.11 Conclusion

Le long de ce chapitre, nous avons exposé les différentes techniques de résolution du problème d'interpolation d'un réseau surfacique, cité en introduction. Nous avons vu l'im-

portance de la représentation polynomiale des facettes, le problème de twist qui l'accompagne et les différentes approches le résolvant. A part des propriétés théoriques, l'aspect pratique des méthodes en vue d'une utilisation dans un système graphique 3D ainsi que la qualité des surfaces résultantes ont été les points importants dans la discussion.

Dans le chapitre suivant, nous allons proposer une méthode d'interpolation qui peut être considérée comme un mélange d'idées, car elle utilise à la fois deux approches : l'approche des courbes frontière pour résoudre le problème de twist et l'approche de subdivision pour baisser le degré des facettes polynomiales et pour fournir plus de paramètres de forme.

Chapitre 3

Méthode de 4-split

3.1 Introduction

Dans ce chapitre nous présentons une nouvelle méthode d'interpolation G^1 d'un réseau surfacique triangulaire, récemment développée par Hahmann et Bonneau [65]. Elle résout le problème de compatibilité de twist d'une manière analogue à celle de Loop (cf. sect. 2.9.1), mais elle est basée sur une nouvelle idée qui consiste à subdiviser en quatre le triangle domaine par les milieux de ces arêtes, d'où le nom de cette méthode. Ce 4-split du triangle domaine a l'avantage de baisser le degré polynomial de la surface de 6 à 5 par rapport à la méthode de Loop tout en gardant un schéma local, ce qui est très important. Les oscillations éventuelles des courbes frontière quand le réseau est interpolé seront réduites.

Ce chapitre est consacré à une description détaillée de la méthode 4-split car elle est à l'origine des travaux de la présente thèse. Cependant, à plusieurs endroits de ce chapitre, des preuves théoriques de certaines propriétés ont aussi contribué à la version originale de cette méthode. Les chapitres 4, 5, et 6 contiennent les principales contributions reposant sur cette méthode. Les chapitres 4 et 5 représentent deux généralisations de la méthode, et le chapitre 6 une étude détaillée des degrés de liberté et de leur utilisation pratique.

La méthode de 4-split prend en entrée un réseau surfacique triangulé de topologie arbitraire, et construit autour de chaque sommet et pour chaque face quatre facettes quintiques en subdivisant le triangle domaine par les milieux de ces arêtes. Les courbes frontière sont construites en correspondance avec le réseau des arêtes de la triangulation par une méthode analogue à celle de Loop. Ce réseau de courbes est compatible avec le problème de twist et permet de calculer explicitement un vecteur twist compatible pour chaque sous-facette. Ces vecteurs twists sont ensuite utilisés pour calculer les rubans des dérivées trans-frontière le long des courbes frontière assurant la continuité G^1 entre les

macro-facettes. La méthode laisse plusieurs paramètres libres, qui permettent de satisfaire quelques propriétés géométriques de la surface.

Il est intéressant de noter que la subdivision du triangle domaine n'intervient pas dans la résolution du problème de twist contrairement à la subdivision de type Clough-Tocher, mais qu'elle apporte plus de flexibilité dans la construction de la surface et permet de réduire les oscillations éventuelles des courbes frontière. De plus elle respecte la forme des faces triangulaires car elle ne subdivise pas les angles des macro-facettes, à l'inverse de la subdivision de type Clough-Tocher. Les paramètres de formes sont disponibles soit lors de la construction des courbes frontière, soit lors du remplissage des facettes par les points de Bézier intérieurs.

3.2 Notations

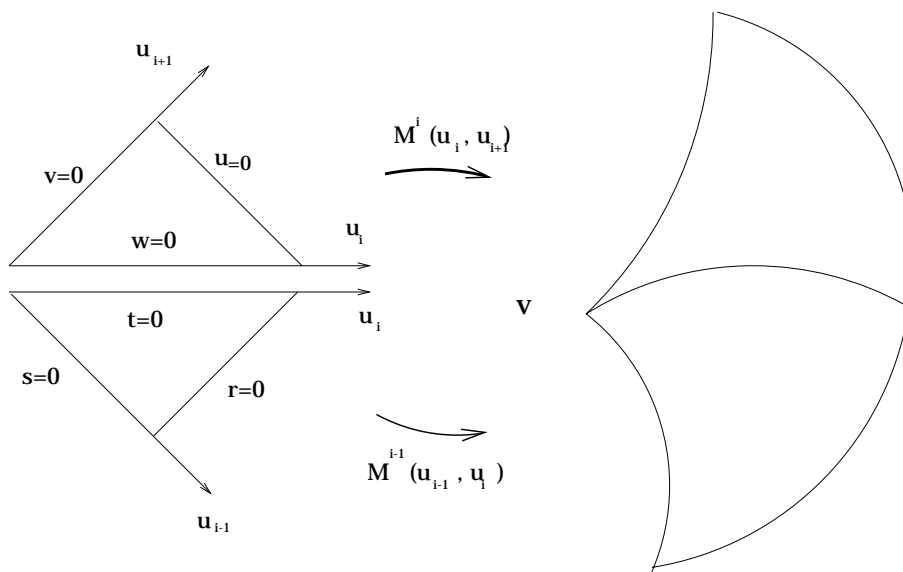


FIG. 3.1 – Paramétrisation de deux facettes voisines.

Les lettres en gras désignent les vecteurs, les matrices et les fonctions vectorielles, tandis que les symboles grecs et les lettres en italiques désignent les scalaires et les fonctions réelles. Une facette triangulaire de Bézier est notée en général par la lettre M . Ses points de contrôle seront notés par la lettre \mathbf{b} . Un indice placé en haut à droite de \mathbf{b} désigne l'indice des facettes autour d'un sommet tandis qu'un indice placé en bas à droite ordonne les points de contrôle.

Considérons un sommet quelconque \mathbf{v} du réseau surfacique et soient M^i , $i = 1, \dots, n$ les n facettes à construire autour de \mathbf{v} (n s'appelle le degré de \mathbf{v}). Considérons deux facettes

voisines M^{i-1} et M^i . La paramétrisation de M^i sur un triangle Δ_i de \mathbb{R}^2 est donné par le repère construit à partir d'un sommet de ce triangle et des deux côtés passant par ce sommet¹, voir fig. 3.1. Un point de Δ_i est défini soit par ses coordonnées euclidiennes (u_i, u_{i+1}) par rapport à ce repère soit par ses coordonnées barycentriques $(\lambda_1, \lambda_2, \lambda_3)$ par rapport à ce triangle. Nous pouvons donc écrire $M^i(u_i, u_{i+1})$ fonction de (u_i, u_{i+1}) comme nous pouvons écrire $M^i(\lambda_1, \lambda_2, \lambda_3)$ fonction de $(\lambda_1, \lambda_2, \lambda_3)$, $\lambda_1 + \lambda_2 + \lambda_3 = 1$, $\lambda_1, \lambda_2, \lambda_3 \geq 0$. Ces deux notations seront différenciées suivant leur contexte d'utilisation.

La paramétrisation de la facette M^{i-1} est choisie de la même manière sur le triangle Δ_{i-1} par (u_{i-1}, u_i) ou par les coordonnées barycentriques par rapport à Δ_{i-1} . Les deux facettes ont une courbe commune correspondant à $M^i(u_i, 0) = M^{i-1}(0, u_i)$ en coordonnées euclidiennes et à $M^i(1-v, v, 0) = M^{i-1}(1-s, s, 0)$, $v, s \in [0, 1]$ en coordonnées barycentriques.

Raccordement G^1

M^{i-1} et M^i se raccordent G^1 le long de leur courbe commune si et seulement s'il existe trois fonctions réelles $\Phi_i(u_i)$, $\mu_i(u_i)$ et $\nu_i(u_i)$ telles que :

$$\begin{aligned} \Phi_i(u_i) [D_{\mathbf{u}_i} M^i](u_i, 0) &= \mu_i(u_i) [D_{\mathbf{u}_{i-1}} M^{i-1}](0, u_i) + \nu_i(u_i) [D_{\mathbf{u}_{i+1}} M^i](u_i, 0), \\ [D_i M^i](u_i, 0) \times [D_{i+1} M^i](u_i, 0) &\neq 0, \\ \mu_i(u_i) \nu_i(u_i) &> 0, \quad \forall u_i \in [0, 1]. \end{aligned} \tag{3.1}$$

Il a été démontré dans Peters [110] que si M^{i-1} et M^i sont polynomiales alors les fonctions Φ_i , μ_i et ν_i sont des polynômes en u_i et que leur degré vérifie :

$$\begin{aligned} \deg(\Phi_i) &\leq \deg \{[D_{i-1} M^{i-1}](0, u_i)\} + \deg \{[D_{i+1} M^i](u_i, 0)\}, \\ \deg(\mu_i) &\leq \deg \{[D_i M^i](u_i, 0)\} + \deg \{[D_{i+1} M^i](u_i, 0)\}, \\ \deg(\nu_i) &\leq \deg \{[D_{i-1} M^{i-1}](0, u_i)\} + \deg \{[D_i M^i](u_i, 0)\}. \end{aligned} \tag{3.2}$$

Les fonctions $\Phi_i(t)$, $\mu_i(t)$ et $\nu_i(t)$ sont appelées les poids du raccordement G^1 entre M^{i-1} et M^i . On dit aussi que le raccordement G^1 est caractérisé par le triplet $[\deg(\Phi_i), \deg(\mu_i), \deg(\nu_i)]$. Par exemple la construction de Loop est un (2,0,0)-raccordement et celle de Peters est un (2,1,1)-raccordement.

La condition $\mu_i \nu_i > 0$ est nécessaire pour préserver l'orientation des deux patches, l'un par rapport à l'autre. En d'autres termes, M^{i-1} et M^i doivent se situer à deux côtés opposés par rapport à leur courbe commune.

1. Il faut noter que ce repère n'est pas orthogonal

La surface finale est complètement déterminée par la définition des points de contrôle des patches, voir sect. 1.2. Ces patches sont en correspondance bijective avec les faces du réseau surfacique et sont appelés macro-patches ou macro-facettes.

3.3 Traitement du problème de twist

La surface G^1 qu'on souhaite construire doit être polynomiale par morceaux. Donc chacun de ses patches doit appartenir à la classe de continuité C^2 (en fait C^∞), ce qui implique que les twists (dérivées secondes mixtes aux sommets) doivent être identiques pour chaque patch en chacun de ses sommets. Il ne suffit donc pas de remplir uniquement les conditions 3.1 de la continuité G^1 mais il faut aussi remplir une condition de compatibilité de twist (voir aussi sect. 2.4). Cette condition peut être exprimée en dérivant la première équation de 3.1 dans la direction du vecteur \mathbf{u}_i , et ensuite en l'évaluant en $u_i = 0$. On obtient le système suivant de n équations linéaires et n vecteurs twists inconnus, appelé système de compatibilité de twist en \mathbf{v} :

$$\begin{aligned} \mu_i(0) [D_{\mathbf{u}_{i-1}} D_{\mathbf{u}_i} M^{i-1}] (0, 0) + \nu_i(0) [D_{\mathbf{u}_i} D_{\mathbf{u}_{i+1}} M^i] (0, 0) &= \Phi'(0) [D_{\mathbf{u}_i} M^i] (0, 0) \\ &+ \Phi_i(0) [D_{\mathbf{u}_i} D_{\mathbf{u}_i} M^i] (0, 0) + \nu'_i(0) [D_{\mathbf{u}_{i+1}} M^i] (0, 0) \\ &+ \mu'_i(0) [D_{\mathbf{u}_{i-1}} M^{i-1}] (0, 0), \end{aligned} \quad (3.3)$$

$$i = 1, \dots, n.$$

En évaluant 3.1 en $u_i = 0$, on obtient des conditions sur les tangentes en \mathbf{v} des courbes frontière. C'est un système linéaire à n équations linéaires et n vecteurs tangents inconnus:

$$\begin{aligned} \Phi_i(0) [D_{\mathbf{u}_i} M^i] (0, 0) &= \mu_i(0) [D_{\mathbf{u}_{i-1}} M^{i-1}] (0, 0) + \nu_i(0) [D_{\mathbf{u}_{i+1}} M^i] (0, 0), \\ &i \in \{1, \dots, n\}. \end{aligned} \quad (3.4)$$

Tous les indices dans 3.3 et 3.4 sont pris modulo n . Le système 3.4 permet de construire les n vecteurs tangents des courbes frontière sous les conditions G^1 .

Dans la suite, nous allons d'abord voir comment choisir les fonctions réelles Φ_i , μ_i et ν_i . L'observation des inégalités 3.2 montre clairement que le degré des facettes triangulaires M^i dépend des degrés des fonctions réelles Φ_i , μ_i et ν_i . Il est donc très important de choisir ces fonctions de plus bas degré possible. La manière la plus économique, comme nous allons le montrer dans la suite, consiste à prendre Φ_i linéaire et μ_i, ν_i constantes, mais ce choix ne permet pas toujours une solution des deux systèmes 3.4 et 3.3.

3.4 Choix de Loop des fonctions réelles Φ_i, μ_i, ν_i

Loop fait un (2,0,0)-raccordement avec $\Phi_i(u_i) = \Phi_0^i B_0^2(u_i) + \frac{1}{2} B_1^2(u_i) + \Phi_1^i B_2^2(u_i)$, et $\mu_i(u_i) = \nu_i(u_i) \equiv \frac{1}{2}$ (voir sect. 2.9.1).

Pour plus de simplification, il considère que toutes les fonctions Φ_i autour de \mathbf{v} ont une même valeur de position et de dérivée en $u_i = 0$, i.e.

$$\begin{cases} \Phi_i(0) = \Phi^0, \\ \Phi_i'(0) = \Phi^1, \end{cases} \quad \forall i \in \{1, \dots, n\},$$

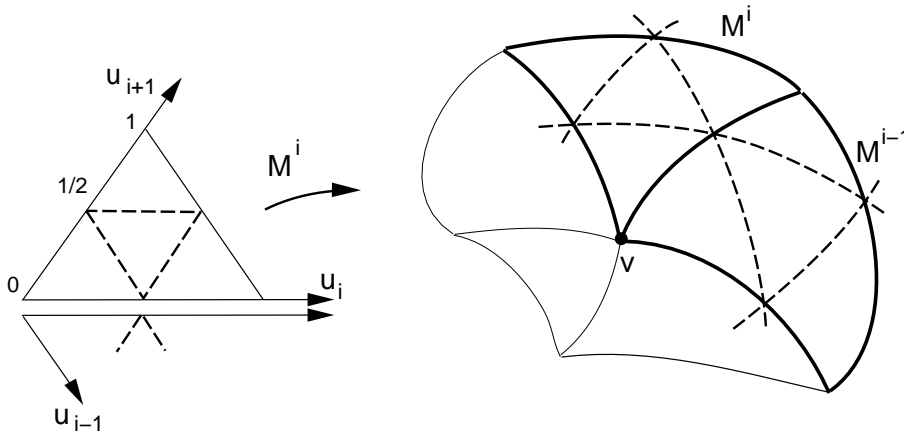
3.4.1 Motivation du choix de Loop

- le choix de $\mu_i = \nu_i \equiv \frac{1}{2}$ assurent une bonne orientation de M^{i-1} et M^{i+1} , l'une par rapport à l'autre, car $\mu_i \nu_i = \frac{1}{4} > 0$.
- le choix de Φ_i quadratique par Loop est motivé par le but de construire un schéma à la fois local et de bas degré. Donc afin de rendre les conditions G^1 au sommet \mathbf{v} indépendantes de celles aux sommets voisins, les valeurs et les dérivées de Φ_i en $u_i = 0$ doivent être indépendantes des sommets voisins. Ainsi prendre Φ_i quadratique se révèle être le choix de plus bas degré laissant les deux conditions G^1 en \mathbf{v} indépendantes des sommets voisins. De plus, le choix du point de contrôle du milieu égal à $\frac{1}{2}$ également permet de baisser le degré de Φ_i , lorsque deux sommets opposés ont le même degré : Φ_i devient linéaire. Si de plus deux sommets ont le degré 6, alors Φ_i devient constante. Cette remarque permet de baisser le degré des facettes si le réseau surfacique présente quelques régularités locales.

Dans la suite, nous allons voir que la méthode de 4-split [65] procède à un raccordement G^1 de type (1,0,0) à l'aide d'une technique de subdivision du triangle domaine en quatre sous-triangles, en insérant de nouveaux sommets aux milieux des arêtes. Cette technique va permettre de réduire le degré polynomial des facettes et d'offrir plus de flexibilité dans la construction des courbes frontière.

3.5 Subdivision du triangle domaine

On subdivise le triangle domaine Δ en joignant les milieux de ses trois côtés. Nous obtenons ainsi quatre sous-triangles, ce qui mène par la suite à la construction de quatre facettes triangulaires par macro-facette, voir fig. 3.2. Ainsi, les courbes frontière et les fonctions réelles vont être polynomiales par morceaux. Cette technique de subdivision est différentes des techniques utilisées dans les autres méthodes d'interpolation. En effet son but n'est pas de résoudre le problème de twist (comme les subdivisions de type Clough-Tocher de la sect. 2.5) mais a pour but d'accomplir l'interpolation des sommets

FIG. 3.2 – *Subdivision du triangle domaine.*

du réseau surfacique et de permettre un contrôle efficace de la forme de la surface finale. On remarque que cette technique engendre de nouveaux sommets de degré 6, aux milieux des arêtes. Donc le problème de twist est aussi posé en ces nouveaux sommets. Mais ce problème sera résolu automatiquement lors de la construction des courbes frontière et des tangentes trans-frontière.

3.6 Choix des fonctions Φ_i , μ_i et ν_i par la méthode du 4-split

On prend $\mu_i = \nu_i \equiv \frac{1}{2}$, Φ_i linéaire par morceaux et on fait les mêmes hypothèses simplificatrices que Loop sur la valeur et la dérivée première de Φ_i , i.e. : $\Phi_i(0) = \Phi^0$ et $\Phi'_i = \Phi^1$, $\forall i \in \{1, \dots, n\}$. Cela veut dire que les valeurs de $\Phi_i(0)$ et $\Phi'_i(0)$ autour de \mathbf{v} sont indépendantes de i .

Afin d'alléger l'écriture des conditions G^1 en \mathbf{v} , commençons d'abord par adopter la notation suivante pour les vecteurs des dérivées premières et secondes en \mathbf{v} des courbes frontière :

$$\begin{aligned}
 \mathbf{r}_i^1 &= [D_{\mathbf{u}_i} M^i](0, 0), \\
 \mathbf{r}_i^2 &= [D_{\mathbf{u}_i} D_{\mathbf{u}_i} M^i](0, 0), \\
 \mathbf{t}_i &= [D_{\mathbf{u}_i} D_{\mathbf{u}_{i+1}} M^i](0, 0) = [D_{\mathbf{u}_{i+1}} D_{\mathbf{u}_i} M^i](0, 0), \\
 & \quad i = 1, \dots, n.
 \end{aligned} \tag{3.5}$$

Les indices sont pris modulo n . On peut vérifier facilement que

$$[D_{\mathbf{u}_{i+1}} M^i](0, 0) = [D_{\mathbf{u}_{i+1}} M^{i+1}](0, 0) = \mathbf{r}_{i+1}^1.$$

Les vecteurs \mathbf{t}_i sont les vecteurs twist, ils doivent être compatibles avec la représentation polynomiale des facettes qu'on va construire autour de \mathbf{v} .

En utilisant ces hypothèses et ces notations, les conditions 3.4 et les conditions de compatibilité de twist 3.3 permettent d'écrire les conditions G^1 suivantes au sommet \mathbf{v} :

$$\Phi^0 \mathbf{r}_i^1 - \frac{1}{2} \mathbf{r}_{i-1}^1 - \frac{1}{2} \mathbf{r}_{i+1}^1 = 0 \quad (3.6)$$

et

$$\frac{1}{2} \mathbf{t}_{i-1} + \frac{1}{2} \mathbf{t}_i = \Phi^1 \mathbf{r}_i^1 + \Phi^0 \mathbf{r}_i^2, \quad (3.7)$$

Les équations 3.6 permettent la construction des tangentes des courbes frontière ainsi que de Φ^0 . Les équations 3.7 sont les conditions de compatibilité de twist, elles permettent de choisir les vecteurs twist \mathbf{t}_i , $i = 1, \dots, n$ qui vont donner une définition compatible des facettes polynomiales. Il est intéressant de noter que le second membre des équations 3.7 dépend seulement des courbes frontière.

La proposition suivante traite les deux systèmes précédents quand le sommet est de degré impair.

Proposition 1 *Soit \mathbf{N} le vecteur normal au sommet \mathbf{v} . Supposons que les vecteurs \mathbf{r}_i^2 sont choisis arbitrairement. Si n est impair ($n > 2$) alors on peut choisir d'une manière arbitraire les vecteurs tangents \mathbf{r}_i^1 dans le plan passant par \mathbf{v} et normal à \mathbf{N} , tels que le système 3.6 soit vérifié pour une valeur bien définie de Φ^0 et pour laquelle le système 3.7 admet une solution unique pour les vecteurs twist.*

Preuve:

La matrice du système 3.6 est

$$\mathbf{M}_r = \begin{pmatrix} \phi^0 & -\frac{1}{2} & & & & -\frac{1}{2} \\ -\frac{1}{2} & \phi^0 & -\frac{1}{2} & & & \\ & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot \\ & & & & -\frac{1}{2} & \phi^0 & -\frac{1}{2} \\ -\frac{1}{2} & & & & -\frac{1}{2} & \phi^0 \end{pmatrix}.$$

Il a été démontré dans le théorème 3.2.2 de Davis [26] que le déterminant de \mathbf{M}_r est égal à $(-1)^n \prod_{j=1}^n (\cos(\frac{2\pi j}{n}) - \Phi^0)$ (voir aussi [84]). Donc pour trouver une solution non triviale du système homogène, on doit annuler ce déterminant. Pour cela, on choisit

$$\Phi^0 = \cos\left(\frac{2\pi k}{n}\right), \quad (3.8)$$

pour un $k \in \{1, \dots, n\}$. D'autre part les équations 3.6 donnent

$$\Phi^0 = \frac{(\mathbf{r}_{i-1}^1 \times \mathbf{r}_{i+1}^1) \cdot \mathbf{N}}{2(\mathbf{r}_i^1 \times \mathbf{r}_{i+1}^1) \cdot \mathbf{N}}. \quad (3.9)$$

Mais Φ^0 est supposée indépendante de i . Donc le membre droit de l'équation 3.9 doit être indépendant de i . Il faut donc trouver les tangentes \mathbf{r}_i^1 d'une manière à ce que Φ^0 soit donnée par 3.8 et 3.9 indépendamment de i . Pour cela, on peut construire les n tangentes dans le plan passant par \mathbf{v} et perpendiculaire à \mathbf{N} telles que chaque tangente est obtenue à partir de la précédente par une rotation autour de \mathbf{N} d'angle $\alpha_n = \frac{2\pi}{n}$. Avec ce choix, on a $(\mathbf{r}_{i-1}^1 \times \mathbf{r}_{i+1}^1) \cdot \mathbf{N} = \sin(2\alpha_n)$ et $(\mathbf{r}_i^1 \times \mathbf{r}_{i+1}^1) \cdot \mathbf{N} = \sin(\alpha_n)$. En remplaçant ces deux valeurs dans 3.9, on obtient

$$\Phi^0 = \frac{\sin 2\alpha_n}{2 \sin \alpha_n} = \cos\left(\frac{2\pi}{n}\right). \quad (3.10)$$

La contrainte 3.8 est également vérifiée par ce choix pour $k = 1$.

Afin de démontrer la deuxième partie de la proposition, commençons par écrire le système 3.7 sous la forme matricielle:

$$\mathbf{M}_t \bar{\mathbf{t}} = \Phi^1 \bar{\mathbf{r}}^1 + \Phi^0 \bar{\mathbf{r}}^2, \quad (3.11)$$

où

$$\mathbf{M}_t = \begin{pmatrix} \frac{1}{2} & & & & & & & & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & & & & & & & \\ & & \cdot & \cdot & & & & & \\ & & & \cdot & \cdot & & & & \\ & & & & \cdot & \cdot & & & \\ & & & & & \cdot & \cdot & & \\ & & & & & & \frac{1}{2} & \frac{1}{2} & \\ & & & & & & & \frac{1}{2} & \frac{1}{2} \end{pmatrix},$$

$$\bar{\mathbf{t}} = \begin{pmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_n \end{pmatrix}, \quad \bar{\mathbf{r}}^1 = \begin{pmatrix} \mathbf{r}_1^1 \\ \vdots \\ \mathbf{r}_n^1 \end{pmatrix} \quad \text{et} \quad \bar{\mathbf{r}}^2 = \begin{pmatrix} \mathbf{r}_1^2 \\ \vdots \\ \mathbf{r}_n^2 \end{pmatrix}.$$

Calculons, à présent, le déterminant de \mathbf{M}_t , en développant suivant la première ligne de

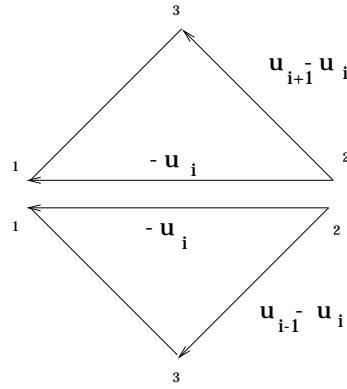
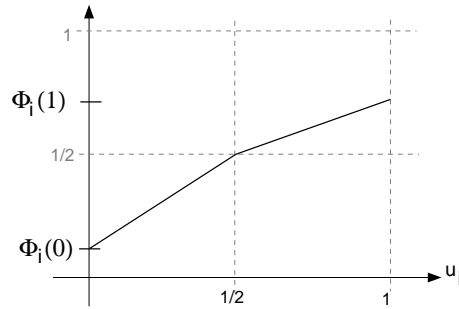


FIG. 3.3 – Directions de dérivation aux sommets opposés.

le système de twist 3.11 dépendra aussi des n_i et le schéma ne sera plus local. Donc pour séparer le calcul des dérivées des Φ_i des valeurs associées aux sommets voisins, le degré des Φ_i devrait être ≥ 2 , comme dans Loop [84], si nous choisissons Φ_i en un seul morceau polynomial.

D'autre part, le degré des facettes est lié au degré des Φ_i par les conditions 3.2. Donc afin d'obtenir une surface de plus bas degré possible, il est souhaitable de trouver Φ_i de degré minimal tout en séparant les calculs autour d'un sommet de ceux autour des sommets voisins.

FIG. 3.4 – la fonction $\Phi_i(u_i)$ se compose de deux morceaux linéaires.

En s'appuyant sur la subdivision du triangle domaine, on peut construire Φ_i **linéaire par morceaux** sur les deux intervalles $[0, \frac{1}{2}]$ et $[\frac{1}{2}, 1]$, avec $\Phi_i(\frac{1}{2}) = \frac{1}{2}$ (voir fig. 3.4), i.e.

$$\Phi_i(u_i) = \begin{cases} \Phi^0(1 - \frac{u_i-0}{1/2-0}) + \frac{1}{2} \frac{u_i-0}{1/2-0}, & \text{si } u_i \in [0, \frac{1}{2}] \\ \frac{1}{2}(1 - \frac{u_i-1/2}{1-1/2}) + \Phi_i(1) \frac{u_i-1/2}{1-1/2}, & \text{si } u_i \in [\frac{1}{2}, 1] \end{cases}$$

d'où

$$\Phi_i(u_i) = \begin{cases} \cos \frac{2\pi}{n}(1 - 2u_i) + u_i, & \text{si } u_i \in [0, \frac{1}{2}] \\ (1 - u_i) + (1 - \cos \frac{2\pi}{n_i})(2u_i - 1), & \text{si } u_i \in [\frac{1}{2}, 1]. \end{cases} \quad (3.12)$$

Le choix de $\Phi_i(\frac{1}{2}) = \frac{1}{2}$ est motivé par le fait que si $n = n_i$, Φ_i devient une seule fonction linéaire, et si de plus $n = n_i = 6$, $\Phi_i \equiv \frac{1}{2}$. Ceci permet de baisser le degré polynomial des facettes lorsque le réseau surfacique présente une régularité locale.

La proposition 1 a montré que si le degré d'un sommet est pair alors le système twist ne donne pas en général une solution du problème de twist autour de ce sommet. Dans la section suivante les courbes frontière seront construites de manière à ce que ce système possède toujours une solution. La subdivision du triangle domaine sera aussi utilisée pour faire une construction de Φ_i d'une manière optimale afin d'obtenir des facettes de degré au plus égal à 5.

3.7 Construction du réseau de courbes frontière

Les courbes frontière sont construites en correspondance avec les arêtes du réseau surfacique en entrée. Cette étape est très cruciale dans la construction de la surface, car la forme des courbes frontière a une grande influence sur la forme de la surface. Les conditions requises sur les courbes frontières sont les suivantes:

- interpolation des sommets du réseau surfacique,
- réalisation des conditions G^1 aux sommets, i.e. les deux systèmes 3.6 et 3.7
- respect de la localité de l'algorithme.

Cette dernière condition implique que les données des courbes frontière utilisées pour la réalisation des conditions G^1 en un sommet sont strictement indépendantes de celles aux sommets opposés. Les données de courbes qui entrent dans les systèmes 3.6 et 3.7 sont les dérivées premières et secondes. Mais on sait que le degré minimal pour séparer les dérivées premières et secondes aux deux extrémités d'une courbe de Bézier est au moins égal à 5. Une autre manière de rendre indépendantes ces données consiste à construire une courbe **polynomiale par morceaux** constituée de deux segments cubiques se joignant avec la continuité C^1 .

Considérons un sommet quelconque \mathbf{v} du réseau surfacique et introduisons la notation suivante

$$\bar{\mathbf{p}} = \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \end{pmatrix},$$

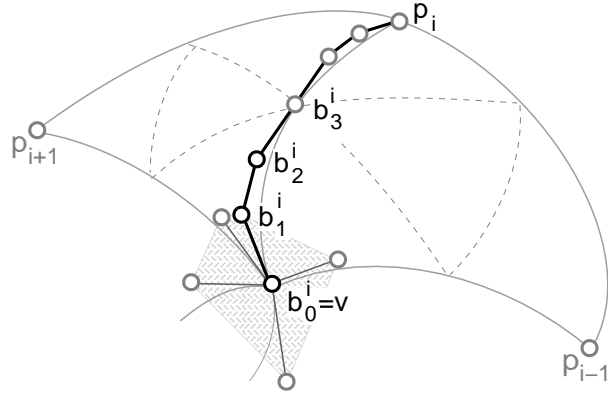


FIG. 3.5 – points de contrôle des courbes frontière autour d'un sommet quelconque \mathbf{v} .

pour le voisinage de \mathbf{v} composé des n sommets \mathbf{p}_i , $i = 1, \dots, n$ de degrés respectifs n_i (n le degré de \mathbf{v}). Les éléments de $\bar{\mathbf{p}}$ sont ordonnés tels que leur indice varie dans le sens trigonométrique autour de \mathbf{v} par rapport à un certain vecteur normal extérieur fixé en \mathbf{v} .

Considérons maintenant la construction de la courbe frontière cubique par morceaux, joignant \mathbf{v} et \mathbf{p}_i . Elle est composée de deux segments cubiques définis respectivement sur les intervalles $[0, \frac{1}{2}]$ et $[\frac{1}{2}, 1]$. Notons le premier segment par son polygone de Bézier $\mathbf{b}_0^i, \dots, \mathbf{b}_3^i$. Le deuxième segment sera construit d'une manière symétrique à partir du sommet \mathbf{p}_i . Les points \mathbf{b}_0^i , \mathbf{b}_1^i et \mathbf{b}_2^i définissent la tangente et la dérivée seconde de la courbe en $u_i = 0$. Ils doivent être construits indépendamment des points \mathbf{p}_i du deuxième segment, afin que les dérivées premières et secondes en chaque sommet soient isolées. Le point de jonction \mathbf{b}_3^i , appelé point milieu, doit assurer un raccordement C^1 entre les deux segments de la courbe. En effet, ce point correspond à la valeur $u_i = \frac{1}{2}$, le milieu de l'arête du triangle domaine dans lequel la subdivision du triangle domaine est opérée. Si on note les points de contrôle du deuxième segment par $\mathbf{b}_0^k, \mathbf{b}_1^k, \mathbf{b}_2^k$ et $\mathbf{b}_3^k = \mathbf{b}_3^{i-2}$ alors \mathbf{b}_3^i est le milieu du segment $[\mathbf{b}_2^i \mathbf{b}_2^k]$.

Pour plus de clarté, les notations matricielles suivantes sont également introduites :

$$\bar{\mathbf{b}}_0 = \begin{pmatrix} \mathbf{b}_0^1 \\ \vdots \\ \mathbf{b}_0^n \end{pmatrix}, \quad \bar{\mathbf{b}}_1 = \begin{pmatrix} \mathbf{b}_1^1 \\ \vdots \\ \mathbf{b}_1^n \end{pmatrix}, \quad \bar{\mathbf{b}}_2 = \begin{pmatrix} \mathbf{b}_2^1 \\ \vdots \\ \mathbf{b}_2^n \end{pmatrix},$$

en rappelant qu'un indice placé en haut à droite varie dans le sens trigonométrique autour du sommet, tandis qu'un indice en bas à droite ordonne les points de contrôle d'une même courbe frontière.

2. k est l'indice de \mathbf{v} dans le voisinage de \mathbf{p}_i .

3.7.1 Construction de $\bar{\mathbf{b}}_0$: Interpolation ou Approximation

Les deux extrémités de chaque courbe frontière sont simplement les deux sommets correspondant, i.e.

$$\mathbf{b}_0^i = \mathbf{v}, \quad \forall i \in \{1, \dots, n\},$$

si l'on désire interpoler les sommets. Mais on peut aussi vouloir approximer les sommets, en construisant \mathbf{b}_0^i par une formule plus général du type

$$\mathbf{b}_0^i = \alpha \mathbf{v} + (1 - \alpha) \frac{1}{n} \sum_{j=1}^n \mathbf{p}_j, \quad (3.13)$$

qui représente un cadre plus général pour \mathbf{b}_0^i car le paramètre α bascule \mathbf{b}_0^i sur le segment joignant \mathbf{v} et le centroïde de $\bar{\mathbf{p}}$. L'interpolation devient un cas particulier de ce cadre plus général, et s'obtient pour $\alpha = 1$. Matriciellement 3.13 peut s'écrire :

$$\bar{\mathbf{b}}_0 = \alpha \bar{\mathbf{v}} + \mathbf{B}^0 \bar{\mathbf{p}}, \quad (3.14)$$

où \mathbf{B}^0 est une matrice $n \times n$ avec

$$\mathbf{B}_{ij}^0 = \frac{1 - \alpha}{n}, \quad i, j = 1, \dots, n.$$

3.7.2 Construction de $\bar{\mathbf{b}}_1$

On doit construire les points \mathbf{b}_1^i dans un même plan qui sera le plan tangent de la surface finale. En effet, ces points définissent les tangentes \mathbf{r}_i^1 des courbes frontière en \mathbf{v} par :

$$\mathbf{r}_i^1 = 6(b_1^i - b_0^i). \quad (3.15)$$

Notons que le coefficient 6 provient du fait que le premier segment d'une courbe frontière est défini sur l'intervalle $[a, b] = [0, \frac{1}{2}]$, donc la dérivée est multipliée par le coefficient $\frac{1}{b-a} = \frac{1}{1/2-0} = 2$.

Le vecteur $\bar{\mathbf{r}}^1 = [\mathbf{r}_1^1, \dots, \mathbf{r}_n^1]^T$ doit satisfaire les équations 3.6 et appartenir à l'espace image de la matrice twist \mathbf{M}_t du système 3.11.

Ces tangentes sont construites à partir de $\mathbf{p}_1, \dots, \mathbf{p}_n$ par

$$\mathbf{r}_i^1 = \sum_{j=1}^n \beta_j \mathbf{p}_{i+j-1} = \beta_1 \mathbf{p}_i + \beta_2 \mathbf{p}_{i+1} + \dots + \beta_n \mathbf{p}_{i-1},$$

où β_j sont n nombres réels utilisés de manière cyclique pour définir toutes les tangentes \mathbf{r}_i^1 . Dans l'expression des \mathbf{r}_i^1 , le facteur β_1 est toujours placé devant \mathbf{p}_i . D'une part, on veut que \mathbf{r}_i^1 soit plus proche de l'arête $[\mathbf{v}\mathbf{p}_i]$ que de toutes les autres arêtes $[\mathbf{v}\mathbf{p}_j]$, $j \neq i$ pour des

raisons géométriques. Ceci pousse à choisir β_1 (coefficient de \mathbf{p}_i) plus grand que tous les autres coefficients β_j pour $j \neq 1$. D'autre part, les \mathbf{r}_i^1 sont soumises aux contraintes 3.6:

$$\Phi^0 \mathbf{r}_i^1 = \frac{1}{2} \mathbf{r}_{i-1}^1 + \frac{1}{2} \mathbf{r}_{i+1}^1,$$

d'où

$$\Phi^0 \beta_i = \frac{1}{2} \beta_{i-1} + \frac{1}{2} \beta_{i+1}. \quad (3.16)$$

On peut choisir la solution suivante:

$$\beta_j = \frac{6\beta}{n} \cos\left(\frac{2\pi(j-i)}{n}\right), \quad j = 1, \dots, n,$$

où β est un paramètre de forme. D'où, d'après Van Wijk [129] (page 6)

$$\mathbf{r}_i^1 = \frac{6\beta}{n} \sum_{j=1}^n \cos\left(\frac{2\pi(j-i)}{n}\right) \mathbf{p}_j. \quad (3.17)$$

Dans la formule des β_j , on divise par n pour que la longueur de \mathbf{r}_i^1 ne grandisse pas linéairement avec le nombre n des sommets voisins. Le paramètre β contrôle les longueurs des \mathbf{r}_i^1 . Remarquons aussi la part prépondérante de l'arête $[\mathbf{v}\mathbf{p}_i]$ dans 3.17: $\beta_1 = \frac{6\beta}{n} > \beta_j$, $\forall j = 2, \dots, n$. En utilisant 3.15 et 3.14 les équations 3.17 s'écrivent matriciellement sous la forme:

$$\bar{\mathbf{b}}^1 = \alpha \bar{\mathbf{v}} + \mathbf{B}^1 \bar{\mathbf{p}} \quad (3.18)$$

où \mathbf{B}^1 est une matrice carrée de dimension $n \times n$, avec

$$\mathbf{B}_{ij}^1 = \frac{1}{n} \left[1 - \alpha + \beta \cos\left(\frac{2\pi(j-i)}{n}\right) \right], \quad i, j = 1, \dots, n.$$

3.7.3 Construction de $\bar{\mathbf{b}}_2$

Voyons maintenant la construction du troisième point de contrôle de la courbe joignant \mathbf{v} à \mathbf{p}_i . Les points \mathbf{b}_2^i sont reliés aux dérivées secondes des courbes frontière par la relation suivante

$$\mathbf{r}_i^2 = 24(b_2^i - 2b_1^i + b_0^i). \quad (3.19)$$

Notons que le coefficient 24 provient du fait que nous dérivant deux fois le premier segment cubique de la courbe, qui est défini sur $[0, \frac{1}{2}]$.

Pour que le deuxième membre du système twist 3.11 appartienne à l'espace image de la matrice \mathbf{M}_t , il suffit de construire les points \mathbf{b}_2^i dans cet espace. Nous avons vu en sect. 2.9.1 que Loop était contraint par le degré 4 des courbes frontière et imposait un

choix de \mathbf{b}_2^i comme combinaison affine en fonction des quatre points \mathbf{v} , \mathbf{p}_{i-1} , \mathbf{p}_i et \mathbf{p}_{i+1} . Plus précisément, Loop choisissait $\mathbf{b}_2^i = \mathbf{d}^i$, où

$$\mathbf{d}^i = \frac{1}{6}(2\mathbf{v} + \mathbf{p}_{i-1} + 2\mathbf{p}_i + \mathbf{p}_{i+1}).$$

Ce choix résulte de la nécessité de construire symétriquement ce point d'une extrémité à l'autre de la courbe frontière. Il satisfait toutes les conditions requises. Néanmoins, ce choix conduit à des oscillations indésirables des courbes frontière lorsque le réseau est interpolé, i.e. lorsque $\alpha = 1$. Précisons par souci d'honnêteté que la méthode de Loop n'a pas été développée pour l'interpolation.

Pour la méthode de 4-split, les courbes frontière sont cubiques par morceaux, et la construction de \mathbf{b}_2^i peut être effectuée plus librement par

$$\mathbf{b}_2^i = \gamma_0 \mathbf{b}_0^i + \gamma_1 \mathbf{b}_1^i + \gamma_2 \mathbf{d}^i = \mathbf{b}_0^i + \gamma_1 (\mathbf{b}_1^i - \mathbf{b}_0^i) + \gamma_2 (\mathbf{d}^i - \mathbf{b}_0^i),$$

où γ_0 , γ_1 et γ_2 sont des paramètres de forme avec $\gamma_0 + \gamma_1 + \gamma_2 = 1$. $\bar{\mathbf{b}}_2$ est une combinaison affine des trois vecteurs $\bar{\mathbf{b}}_0$, $\bar{\mathbf{b}}_1$ et $\bar{\mathbf{d}}$ appartenant à l'espace image de \mathbf{M}_t , il appartient donc aussi à l'espace image de \mathbf{M}_t . Matriciellement, cela donne :

$$\bar{\mathbf{b}}^2 = [(\gamma_0 + \gamma_1)\alpha + \frac{\gamma_2}{3}] \bar{\mathbf{v}} + \mathbf{B}^2 \bar{\mathbf{p}} \quad (3.20)$$

où \mathbf{B}^2 est la matrice $n \times n$ suivante:

$$\mathbf{B}_{ij}^2 = \frac{1}{n} \left[(\gamma_0 + \gamma_1)(1 - \alpha) + \gamma_1 \beta \cos\left(\frac{2\pi(j-i)}{n}\right) \right] + \gamma_2 \begin{cases} 1/6, & \text{si } j = i - 1, i + 1 \\ 1/3, & \text{si } j = i \\ 0, & \text{sinon.} \end{cases}$$

3.7.4 Construction de $\bar{\mathbf{b}}_3$

Le point \mathbf{b}_3^i doit appartenir à la courbe, car il est le point de jonction entre les deux segments cubiques de la courbe frontière. Il doit être construit afin d'obtenir une continuité C^1 entre les deux segments. Il est donc le milieu du segment joignant \mathbf{b}_2^i et \mathbf{b}_2^k , où k est l'indice de \mathbf{v} dans le voisinage de \mathbf{p}_i .

Maintenant que nous avons construit les courbes frontière telles que les vecteurs $\bar{\mathbf{r}}^1$ et $\bar{\mathbf{r}}^2$ sont dans l'espace image de \mathbf{M}_t , nous allons voir dans le paragraphe suivant comment résoudre le système 3.11 pour les vecteurs twist.

3.8 Calcul du vecteur twist compatible

Avec la construction de la section 3.7 des courbes frontières, le système twist 3.11 admet des solutions. On veut déterminer explicitement une solution de ce système. Autrement dit nous allons chercher un vecteur $\bar{\mathbf{t}} = (\mathbf{t}_i)$ de $\mathbb{R}^{n \times 3}$ compatible avec la représentation

polynomiale des n facettes M^i autour de \mathbf{v} .

Pour cela et puisque $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ appartiennent à l'espace image de \mathbf{M}_t , nous allons d'abord trouver trois vecteurs $\tilde{\mathbf{b}}_0, \tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2 \in \mathbb{R}^{n \times 3}$ tels que :

$$\mathbf{M}_t \tilde{\mathbf{b}}_0 = \bar{\mathbf{b}}_0, \quad \mathbf{M}_t \tilde{\mathbf{b}}_1 = \bar{\mathbf{b}}_1, \quad \mathbf{M}_t \tilde{\mathbf{b}}_2 = \bar{\mathbf{b}}_2.$$

Pour pouvoir trouver ces trois vecteurs, on doit d'abord caractériser l'espace image de la matrice \mathbf{M}_t . À présent nous donnerons les formules directes de ces vecteurs dans le théorème suivant, sans trop de détail qui sera laissé au chapitre 4.

Théorème 1 *Pour tout n , les trois vecteurs $\tilde{\mathbf{b}}_0, \tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2$ sont donnés par*

$$\begin{aligned} \tilde{\mathbf{b}}_0^i &= \mathbf{b}_0^i, \\ \tilde{\mathbf{b}}_1^i &= \alpha \bar{\mathbf{v}} + \frac{1}{n} \sum_{j=1}^n \left[(1 - \alpha) + \beta \left(\cos \frac{2\pi(j-i)}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi(j-i)}{n} \right) \right] \mathbf{p}_j, \\ \tilde{\mathbf{b}}_2^i &= \gamma_0 \mathbf{b}_0^i + \gamma_1 \tilde{\mathbf{b}}_1^i + \gamma_2 \left(\frac{1}{3} \mathbf{v} + \frac{1}{3} \mathbf{p}_i + \frac{1}{3} \mathbf{p}_{i+1} \right), \end{aligned}$$

Preuve:

Pour la construction de $\tilde{\mathbf{b}}_0$ et $\tilde{\mathbf{b}}_1$ tels que $\mathbf{M}_t \tilde{\mathbf{b}}_0 = \bar{\mathbf{b}}_0$ et $\mathbf{M}_t \tilde{\mathbf{b}}_1 = \bar{\mathbf{b}}_1$, voir [84]. Pour la construction de $\tilde{\mathbf{b}}_2$ tel que $\mathbf{M}_t \tilde{\mathbf{b}}_2 = \bar{\mathbf{b}}_2$, voir [65]. \square

Les constructions faites dans [84, 65] sont basées sur une interprétation géométrique simple de l'espace image de \mathbf{M}_t :

si \mathbf{V} est un vecteur de \mathbf{M}_t , alors il existe $\bar{\mathbf{V}} \in \mathbb{R}^{n \times 3}$ tel que $\mathbf{V} = \mathbf{M}_t \bar{\mathbf{V}}$ et les n points de \mathbf{V} se trouvent sur les milieux des cotés successifs d'un polygone dont les sommets sont les n points de $\bar{\mathbf{V}}$, voir fig. 3.6.

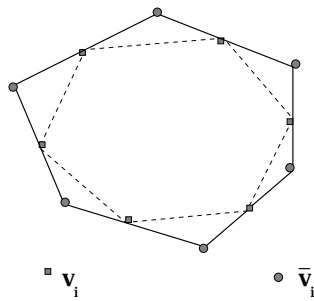


FIG. 3.6 – $\mathbf{M}_t \bar{\mathbf{V}} = \mathbf{V}$ signifie que les points \mathbf{V}_i sont les milieux des cotés successifs du polygone de sommets $\bar{\mathbf{V}}_i$.

En utilisant ce théorème et les deux équations 3.15 et 3.19, le deuxième membre de 3.11 s'écrit :

$$\begin{aligned}
\Phi^1 \mathbf{r}_i^1 + \Phi^0 \mathbf{r}_i^2 &= 6\Phi^1(\mathbf{b}_1^i - \mathbf{b}_0^i) + 24\Phi^0(\mathbf{b}_2^i - 2\mathbf{b}_1^i + \mathbf{b}_0^i) \\
&= (-6\Phi^1 + 24\Phi^0)\mathbf{b}_0^i + (6\Phi^1 - 48\Phi^0)\mathbf{b}_1^i + 24\Phi^0\mathbf{b}_2^i \\
&= \mathbf{M}_t \left[(-6\Phi^1 + 24(1 + \gamma_0)\Phi^0)\mathbf{b}_0^i + (6\Phi^1 + (-48 + 24\gamma_1)\Phi^0)\tilde{\mathbf{b}}_1^i \right. \\
&\quad \left. + 8\gamma_2\Phi^0(\mathbf{v} + \mathbf{p}_i + \mathbf{p}_{i+1}) \right].
\end{aligned}$$

Une solution explicite du système pour les vecteurs twists peut donc être trouvée :

$$\begin{aligned}
\mathbf{t}_i &= 8\gamma_2\Phi^0(1 - 3\alpha)\mathbf{v} + 8\gamma_2\Phi^0(\mathbf{p}_i + \mathbf{p}_{i+1}) \\
&+ \frac{1}{n} \sum_{j=1}^n \left[-24\gamma_2\Phi^0(1 - \alpha) + \beta(6\Phi^1 + (24\gamma_1 - 48)\Phi^0) \left(\cos \frac{2\pi(j-i)}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi(j-i)}{n} \right) \right] \mathbf{p}_j, \\
&\hspace{25em} i = 1, \dots, n.
\end{aligned} \tag{3.21}$$

Rappelons que $\alpha, \beta, \gamma_1, \gamma_2$ ($\gamma_0 = 1 - \gamma_1 - \gamma_2$) sont des paramètres de forme. α permet de basculer entre l'interpolation et l'approximation, β contrôle les longueurs des tangentes des courbes autour du sommet, et γ_1, γ_2 contrôlent les dérivées secondes.

3.9 Les dérivées trans-frontière

Une fois les courbes frontière trouvées et les vecteurs de twist compatibles calculés, l'étape suivante vers la construction de la surface consiste à construire les rubans de tangence, i.e. les dérivées $D_{\mathbf{u}_{i-1}}M^{i-1}(0, u_i)$ et $D_{\mathbf{u}_{i+1}}M^i(u_i, 0)$ le long de chaque courbe frontière. Elles doivent satisfaire trois conditions :

- condition 3.1 et 3.7 du raccordement G^1 ,
- interpolation des valeurs de twists aux points extrémité de la courbe correspondante,
- être consistantes avec les dérivées des courbes frontière aux extrémités, voir 3.7.

Les valeurs des tangentes trans-frontière aux sommets sont imposées par les courbes frontière elles mêmes, voir fig 3.7. Une manière de satisfaire les conditions G^1 est de définir les rubans de tangences de la manière symétrique suivante :

$$\begin{cases} [D_{\mathbf{u}_{i+1}}M^i](u_i, 0) &= \Phi_i(u_i)[D_{\mathbf{u}_i}M^i](u_i, 0) + \Psi_i(u_i)\mathbf{V}_i(u_i), \\ [D_{\mathbf{u}_{i-1}}M^{i-1}](0, u_i) &= \Phi_i(u_i)[D_{\mathbf{u}_i}M^i](u_i, 0) - \Psi_i(u_i)\mathbf{V}_i(u_i), \end{cases} \tag{3.22}$$

où Ψ_i est une fonction réelle et \mathbf{V}_i est une fonction vectorielle. Il est intéressant de noter que les conditions 3.22 sont des conditions suffisantes pour satisfaire les condition G^1 de 3.1. Pour le voir, il suffit d'additionner les deux équations de 3.22. Il faut maintenant construire Ψ_i et \mathbf{V}_i avec le plus bas degré possible de manière à interpoler les tangentes

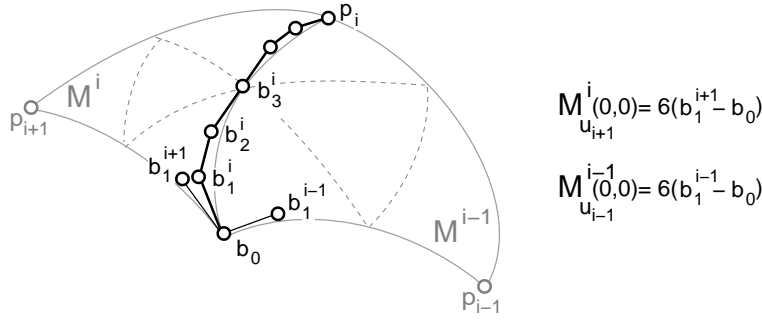


FIG. 3.7 – les tangentes trans-frontière aux points extrémité sont bien fixées par les courbes frontière.

\mathbf{r}_i^1 et les vecteurs de twist \mathbf{t}_i de la section précédente. Les considérations suivantes vont montrer que ψ_i pourra être prise linéaire et \mathbf{V}_i quadratique par morceaux.

Le produit $\Phi_i [D_{\mathbf{u}}, M^i]$ est cubique par morceaux. Donc il faut prendre Ψ_i et \mathbf{V}_i telles que le degré polynomial de leur produit soit inférieur ou égal à trois. En tenant compte de la subdivision et en suivant Loop [84], les valeurs de $\Psi_i(0)$ et de $\mathbf{V}_i(0)$ sont déterminées à partir de la première équation de 3.22 évaluée en $u_i = 0$, i.e. ;

$$\begin{aligned} [D_{\mathbf{u}_{i+1}} M^i] (0, 0) &= \Phi_i(0) [D_{\mathbf{u}}, M^i] (0, 0) + \Psi_i(0) \mathbf{V}_i(0), \\ &= \Phi^0 \mathbf{r}_i^1 + \Psi_i(0) \mathbf{V}_i(0), \end{aligned}$$

mais $[D_{\mathbf{u}_{i+1}} M^i] (0, 0) = [D_{\mathbf{u}_{i+1}} M^{i+1}] (0, 0) = \mathbf{r}_{i+1}^1$, d'où

$$\mathbf{r}_{i+1}^1 = \Phi^0 \mathbf{r}_i^1 + \Psi_i(0) \mathbf{V}_i(0). \quad (3.23)$$

En faisant l'hypothèse simplificatrice $\Psi_i(0) = \Psi^0$ (indépendamment de i) et en remplaçant les équations 3.10, 3.15 et 3.19 dans l'équation 3.23, on obtient :

$$\Psi^0 \mathbf{V}_i(0) = \sin \frac{2\pi}{n} \cdot \sum_{j=1}^n \frac{6\beta}{n} \sin \frac{2\pi(j-i)}{n} \mathbf{p}_j,$$

Un choix approprié pour Ψ^0 est alors $\Psi^0 = \sin \frac{2\pi}{n}$, d'où

$$\mathbf{V}_i(0) = \frac{6\beta}{n} \sum_{j=1}^n \sin \frac{2\pi(j-i)}{n} \mathbf{p}_j. \quad (3.24)$$

La valeur de $\Psi_i(1) \mathbf{V}_i(1)$ peut être obtenue d'une manière analogue à partir du sommet \mathbf{p}_i . En particulier $\Psi_i(1) = \sin \frac{2\pi}{n_i}$, où n_i est le degré du sommet \mathbf{p}_i . Ψ_i est donc la fonction linéaire suivante :

$$\Psi_i(u_i) = \sin \frac{2\pi}{n} (1 - u_i) + \sin \frac{2\pi}{n_i} u_i, \quad u_i \in [0, 1]. \quad (3.25)$$

En dérivant la première équation de 3.22 dans la direction de \mathbf{u}_i , puis en évaluant le résultat en $u_i = 0$, le système de twist est obtenu sous la forme :

$$\begin{aligned} [D_{\mathbf{u}_{i+1}} D_{\mathbf{u}_i} M^i](0,0) &= \Phi'_i(0) [D_{\mathbf{u}_i} M^i](0,0) + \Phi_i(0) [D_{\mathbf{u}_i} D_{\mathbf{u}_i} M^i](0,0) + \Psi'_i(0) \mathbf{V}_i(0) + \\ &\quad + \Psi_i(0) \mathbf{V}'_i(0). \end{aligned} \quad (3.26)$$

On voit apparaître $\mathbf{V}'_i(0)$ dans une relation linéaire avec le vecteur twist

$$\mathbf{t}_i = D_{\mathbf{u}_{i+1}} D_{\mathbf{u}_i} M^i(0,0) = D_{\mathbf{u}_i} D_{\mathbf{u}_{i+1}} M^i(0,0).$$

On en déduit que

$$\mathbf{V}'_i(0) = \frac{1}{\Psi_0} [\mathbf{t}_i - \Phi^1 \mathbf{r}_i^1 - \Phi^0 \mathbf{r}_i^2 - \Psi_i^1 \mathbf{V}_i(0)], \quad (3.27)$$

avec $\Psi_i^1 := \Psi'_i(0) = \sin \frac{2\pi}{n_i} - \sin \frac{2\pi}{n}$ (dépendant de n_i).

La valeur de la dérivée $\mathbf{V}'_i(1)$ est calculée de la même manière à partir du sommet opposé \mathbf{p}_i . L'interpolation d'Hermite cubique permettrait l'interpolation des valeurs $\mathbf{V}_i(0), \mathbf{V}_i(1), \mathbf{V}'_i(0), \mathbf{V}'_i(1)$. Mais la subdivision adoptée permet de prendre \mathbf{V}_i comme une fonction quadratique par morceaux, avec une continuité C^1 au point de jonction. Les points de Bézier du premier segment de \mathbf{V}_i sont donnés par

$$\begin{aligned} \mathbf{v}_0^i &= \sum_{j=1}^n \frac{6\beta}{n} \sin \frac{2\pi(j-i)}{n} \mathbf{p}_j \\ \mathbf{v}_1^i &= \frac{4}{\Psi_0^2} \gamma_2 \Phi^0 (\mathbf{p}_{i+1} - \mathbf{p}_{i-1}) \\ &\quad + \sum_{j=1}^n \frac{1}{\Psi_0^2} [(6\Phi^1 - 48\Phi^0 + 24\gamma_1 \Phi^0) \tan \frac{\pi}{n} - 6\Psi_i^1] \frac{\beta}{n} \sin \frac{2\pi(j-i)}{n} \mathbf{p}_j \end{aligned} \quad (3.28)$$

\mathbf{v}_2^i doit être choisi sous la seule contrainte $\mathbf{V}_i(\frac{1}{2}^-) = \mathbf{V}_i(\frac{1}{2}^+)$.

Remarque 6 Comme nous allons voir dans la sect. 3.10.1, la seule contrainte sur $\mathbf{V}_i(u_i)$ en $u_i = \frac{1}{2}$ est la continuité C^0 . Néanmoins, nous prendrons \mathbf{V}_i de continuité C^1 , i.e. \mathbf{v}_2^i sera choisi au milieu du segment joignant $[\mathbf{v}_1^i \mathbf{v}_1^k]$, où \mathbf{v}_1^k est obtenu à partir du sommet opposé \mathbf{p}_i , avec k l'indice de \mathbf{v} dans le voisinage de \mathbf{p}_i .

Remarque 7 Les rubans de tangence sont construits sous forme cubique par morceaux ainsi que les courbes frontière. Ceci laisse penser que les sous-facettes pourraient être quartiques. Mais, comme nous le constaterons en sect. 3.10.1, ceci est en fait impossible en général. Ces sous-facettes devront ainsi être de degré 5.

3.10 Macro-facettes sous la forme de Bézier

Dorénavant, les macro-facettes sont considérées individuellement. La subdivision du domaine mène à la construction de quatre facettes triangulaires par macro-facette pour lesquelles nous donnerons les valeurs des points de contrôle dans cette section et dans l'annexe A. La rangée frontière et la première rangée intérieure des points de contrôle de la macro-facette sont obtenues à partir des courbes frontières (sect. 3.7) et des tangentes trans-frontières (sect. 3.9). Ces points assurent un raccordement G^1 entre les macro-facettes voisines. Pour les formules exactes de ces points, voir annexe A. Pour avoir une surface globale visuellement lisse, les points de contrôle intérieurs restants sont utilisés pour réaliser un raccordement de continuité C^1 entre les quatre sous-facettes. Six points de contrôle par macro-facette restent libres pour un contrôle supplémentaire de sa forme.

3.10.1 Calcul des points de contrôle intérieurs des macro-facettes

Tous les points de contrôle assurant un raccordement G^1 entre les macro-facettes deux à deux voisines sont montrés en couleur noir dans la figure 3.8. Cette section montrera qu'il est possible de raccorder les sous-facettes S^1 , S^2 , S^3 , S^m par un raccordement C^1 à travers les trois courbes internes et expliquera comment utiliser les points de contrôle restants pour atteindre ce but. Mais avant d'aborder cette question, nous allons d'abord

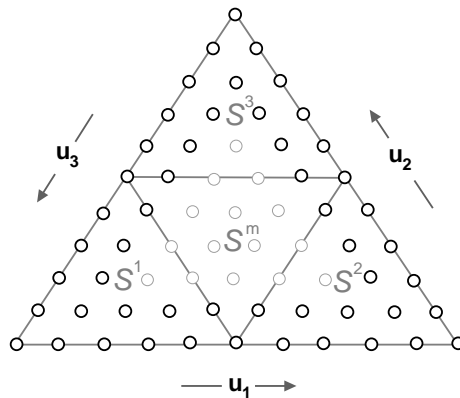


FIG. 3.8 – *points de contrôle provenant des courbes frontière et des tangentes trans-frontière assurant un raccordement G^1 entre les macro-facettes*

traiter les sommets introduits par la subdivision. Ces nouveaux sommets correspondent aux milieux des arêtes du triangle domaine. Ils sont de degré 6, donc le problème de twist est posé en ces sommets. Nous montrerons que ce problème est implicitement résolu par la construction des courbes frontière et des tangentes trans-frontière effectuée dans les sections précédentes.

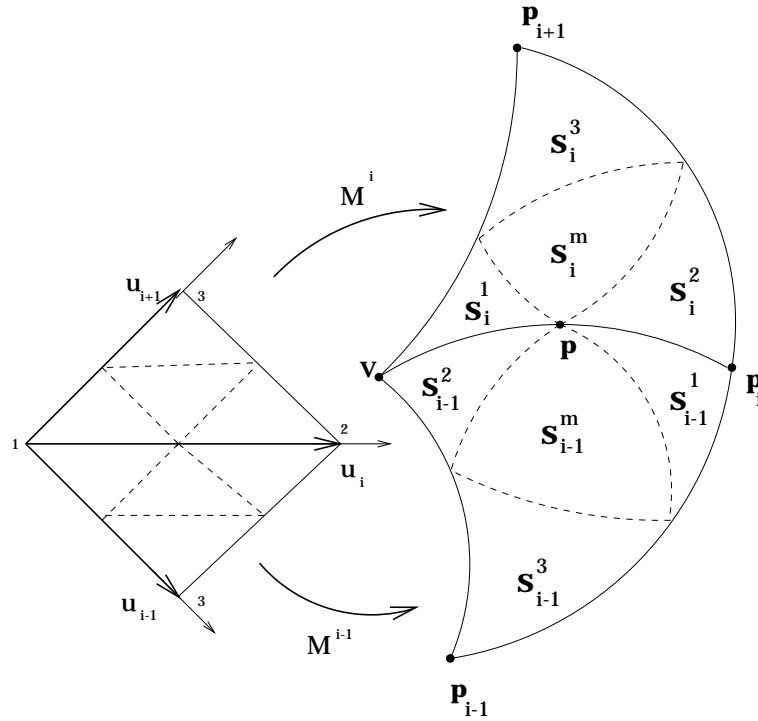


FIG. 3.9 – continuité C^1 au point de subdivision entre les 6 sous-facettes de deux macro-facettes voisines.

Théorème 2 Soit \mathbf{p} le point milieu d'une courbe frontière (i.e. le point de jonction de ces deux segments cubiques). Supposons que cette courbe soit partagée par deux facettes M^{i-1} et M^i , voir fig. 3.9. Alors les six facettes $S_i^1, S_i^m, S_i^2, S_{i-1}^1, S_{i-1}^m, S_{i-1}^2$ se raccordent avec une continuité C^1 en \mathbf{p} .

Pour démontrer ce théorème, nous utilisons le lemme suivant :

Lemme 1 Soit Ω un ouvert de \mathbb{R}^2 contenant l'origine \mathbf{O} et soit $f(u, v)$ une fonction numérique défini sur Ω . S'il existe deux directions indépendantes \mathbf{a} et \mathbf{b} dans \mathbb{R}^2 telles que $D_{\mathbf{a}}f$ et $D_{\mathbf{b}}f$ sont continues au voisinage de \mathbf{O} , alors f est continûment différentiable en \mathbf{O} .

Preuve:

Soit $\epsilon > 0$, tel que le disque ouvert $D(\mathbf{O}, \epsilon)$ de centre \mathbf{O} et de rayon ϵ est inclut dans Ω et soit \mathbf{d} une direction quelconque dans \mathbb{R}^2 . Comme \mathbf{a} et \mathbf{b} sont linéairement indépendants, \mathbf{d} s'écrit de façon unique sous la forme $\mathbf{d} = x\mathbf{a} + y\mathbf{b}$, avec $x, y \in \mathbb{R}$. Dans $D(\mathbf{O}, \epsilon)$, nous

pouvons écrire

$$D_{\mathbf{d}}f = xD_{\mathbf{a}}f + yD_{\mathbf{b}}f,$$

d'où $D_{\mathbf{d}}f$ est continue sur $D(\mathbf{O}, \epsilon)$. Donc f est C^1 en \mathbf{O} .

□

Preuve du théorème 2

Considérons la facette $M(u_i, u_{i+1})$ composée des deux facettes M^{i-1} et M^i , paramétrée sur le carré composé des deux triangles domaines Δ_{i-1} et Δ_i suivant le repère affine $(\mathbf{a}_1, \mathbf{u}_i, \mathbf{u}_{i+1})$. Donc M est définie par

$$M(u_i, u_{i+1}) = \begin{cases} M^{i-1}(u_i + u_{i+1}, -u_{i+1}), & \text{si } (u_i, u_{i+1}) \in \Delta_{i-1}, \\ M^i(u_i, u_{i+1}), & \text{si } (u_i, u_{i+1}) \in \Delta_i. \end{cases}$$

Il faut noter que la reparamétrisation de M^{i-1} avec (u_i, u_{i+1}) est obtenue par le changement du repère $(\mathbf{a}_1, \mathbf{u}_{i-1}, \mathbf{u}_i)$ au repère $(\mathbf{a}_1, \mathbf{u}_i, \mathbf{u}_{i+1})$ avec

$$\mathbf{u}_{i-1} = \mathbf{u}_i - \mathbf{u}_{i+1}.$$

D'après le lemme 1, il suffit de démontrer que les dérivées directionnelles $D_{\mathbf{u}_{i-1}}M$ et $D_{\mathbf{u}_{i+1}}M$ sont continues dans un voisinage de $\mathbf{p} = M(\frac{1}{2}, 0)$. Mais comme les 6 sous-facettes autour de \mathbf{p} sont polynomiales, il suffit alors de démontrer la continuité seulement suivant les trois lignes du domaine passant par $(\frac{1}{2}, 0)$. Pour cela il suffit de démontrer que

$$\begin{aligned} D_{\mathbf{u}_{i+1}}M(\frac{1}{2}^+, 0^+) &= D_{\mathbf{u}_{i+1}}M(\frac{1}{2}^-, 0^+), \\ D_{\mathbf{u}_{i-1}}M(\frac{1}{2}^+, 0^-) &= D_{\mathbf{u}_{i-1}}M(\frac{1}{2}^-, 0^-), \end{aligned} \quad (3.29)$$

et

$$\begin{aligned} D_{\mathbf{u}_{i+1}}M(\frac{1}{2}, 0^+) &= D_{\mathbf{u}_{i+1}}M(\frac{1}{2}, 0^-), \\ D_{\mathbf{u}_{i-1}}M(\frac{1}{2}, 0^+) &= D_{\mathbf{u}_{i-1}}M(\frac{1}{2}, 0^-). \end{aligned} \quad (3.30)$$

Nous pouvons écrire ces conditions en fonction des paramétrisations de M^{i-1} et M^i de la manière suivante: (cf fig. 3.10)

$$\begin{cases} D_{\mathbf{u}_{i+1}}M^i(\frac{1}{2}^+, 0) = D_{\mathbf{u}_{i+1}}M^i(\frac{1}{2}^-, 0), \\ D_{\mathbf{u}_{i-1}}M^{i-1}(0, \frac{1}{2}^+) = D_{\mathbf{u}_{i-1}}M^{i-1}(0, \frac{1}{2}^-), \end{cases} \quad (3.31)$$

et

$$\begin{cases} D_{\mathbf{u}_{i+1}}M^i(\frac{1}{2}, 0) = D_{\mathbf{u}_{i+1}}M^{i-1}(0, \frac{1}{2}), \\ D_{\mathbf{u}_{i-1}}M^i(\frac{1}{2}, 0) = D_{\mathbf{u}_{i-1}}M^{i-1}(0, \frac{1}{2}). \end{cases} \quad (3.32)$$

Il est important de noter que les autres conditions sont automatiquement vérifiées grâce à la continuité C^1 des courbes frontière. Les deux conditions 3.31 sont relatives à une seule macro-facette, elles peuvent être démontrées par l'équation 3.22, en exprimant $D_{\mathbf{u}_{i+1}}M^i$ et $D_{\mathbf{u}_{i-1}}M^{i-1}$ en fonction de Φ_i , $D_{\mathbf{u}_{i+1}}M^i$, Ψ_i et \mathbf{V}_i qui sont toutes continues en $u_i = \frac{1}{2}$.

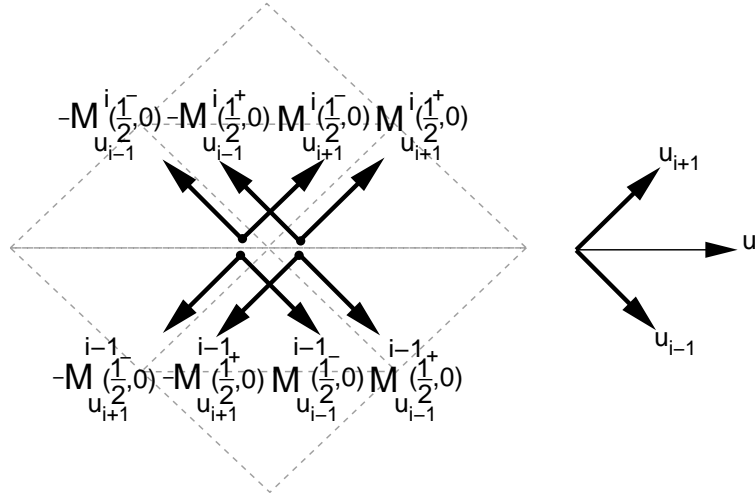


FIG. 3.10 – Les dérivées directionnelles aux points de subdivision, vérifiant les contraintes C^1 .

Pour les deux équations de 3.32 entre les dérivées provenant des deux facettes voisines, nous allons démontrer seulement la première car la deuxième se démontre de manière analogue.

Comme $\mathbf{u}_{i+1} = \mathbf{u}_i - \mathbf{u}_{i-1}$ et d'après 3.22, on vérifie aisément que

$$[D_{\mathbf{u}_{i+1}} M^{i-1}] \left(0, \frac{1}{2}\right) = [D_{\mathbf{u}_i} M^{i-1}] \left(0, \frac{1}{2}\right) - \left\{ \phi\left(\frac{1}{2}\right) [D_{\mathbf{u}_i} M^i] \left(\frac{1}{2}, 0\right) - \Psi_i\left(\frac{1}{2}\right) \mathbf{V}_i\left(\frac{1}{2}\right) \right\}.$$

Mais $\phi\left(\frac{1}{2}\right) = \frac{1}{2}$ et $[D_{\mathbf{u}_i} M^{i-1}] \left(0, \frac{1}{2}\right) = [D_{\mathbf{u}_i} M^i] \left(\frac{1}{2}, 0\right)$. On en déduit que

$$[D_{\mathbf{u}_{i+1}} M^{i-1}] \left(0, \frac{1}{2}\right) = \phi\left(\frac{1}{2}\right) [D_{\mathbf{u}_i} M^i] \left(\frac{1}{2}, 0\right) + \Psi_i\left(\frac{1}{2}\right) \mathbf{V}_i\left(\frac{1}{2}\right) = [D_{\mathbf{u}_{i+1}} M^i] \left(\frac{1}{2}, 0\right).$$

D'où la continuité C^1 en \mathbf{p} . □

Corollaire 1 *Le polygone constitué des six points de contrôle autour de $\mathbf{p} = M^i\left(\frac{1}{2}, 0\right)$ est une carte affine d'un polygone régulier de six cotés, voir fig. 3.11.*

Preuve:

La preuve se déduit de la section 1.2.5 et de la figure 1.7. □

Remarque 8 *Lorsqu'on construit les 4 sous-facettes d'une macro-facette se raccordant G^1 le long des courbes internes, on utilise la méthode de Farin de la section 1.3.3. Avec le corollaire précédent, cette méthode revient à imposer les conditions de la continuité C^1 . De plus, à cause de la subdivision uniforme du triangle domaine, ces conditions sont les conditions de parallélogrammes de la section 1.2.5, voir fig. 3.12. Donc les 4 sous-facettes sont de raccordement C^1 avec la méthode de Farin, voir aussi section 1.2.5.*

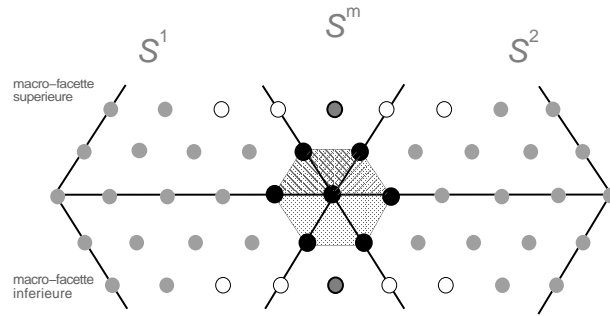


FIG. 3.11 – Les points de contrôle autour d'un point de subdivision forment une carte affine d'un polygone régulier.

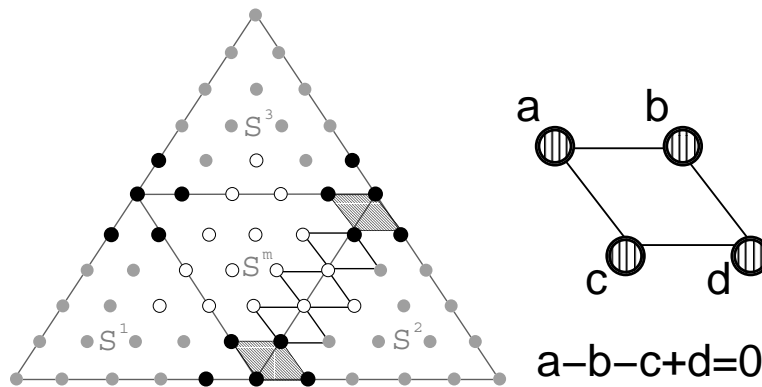


FIG. 3.12 – Conditions de parallélogrammes pour le raccordement C^1 entre les quatre sous-facettes le long des trois courbes internes d'une macro-facettes.

3.10.2 Construction des points de contrôle inconnus

Les conditions de la continuité C^1 entre les sous-facettes internes d'une macro-facette sont montrées fig. 3.12. Toute paire de triangles le long d'une courbe interne doit constituer un parallélogramme, voir la fin de la sect. 1.2.5 et pour plus de détail voir [43, 15].

D'après la remarque 8, la première et la dernière paire de triangles adjacents le long de chaque courbe (voir la figure 3.12) forment déjà deux parallélogrammes. Reste à calculer les points de Bézier libres internes pour que les trois paires de triangles restantes le long de chaque courbe interne constituent aussi des parallélogrammes. Cela peut se faire en quatre étapes:

- (a) choisir d'une façon arbitraire les trois points twists de la facette centrale, ce sont des paramètres libres de forme (voir fig. 3.13.a),
- (b) calculer le troisième et le quatrième point de Bézier le long de chaque courbe interne en utilisant la seconde et la quatrième condition de parallélogramme (voir fig. 3.13.b),

- (c) choisir arbitrairement les trois points de Bézier inconnus de la facette centrale, ils sont aussi des paramètres de forme libres (voir fig. 3.13.c),
- (d) calculer les trois derniers points inconnus de Bézier restants des facettes extérieures, en utilisant la troisième condition de parallélogramme le long de chaque courbe (voir fig. 3.13.d).

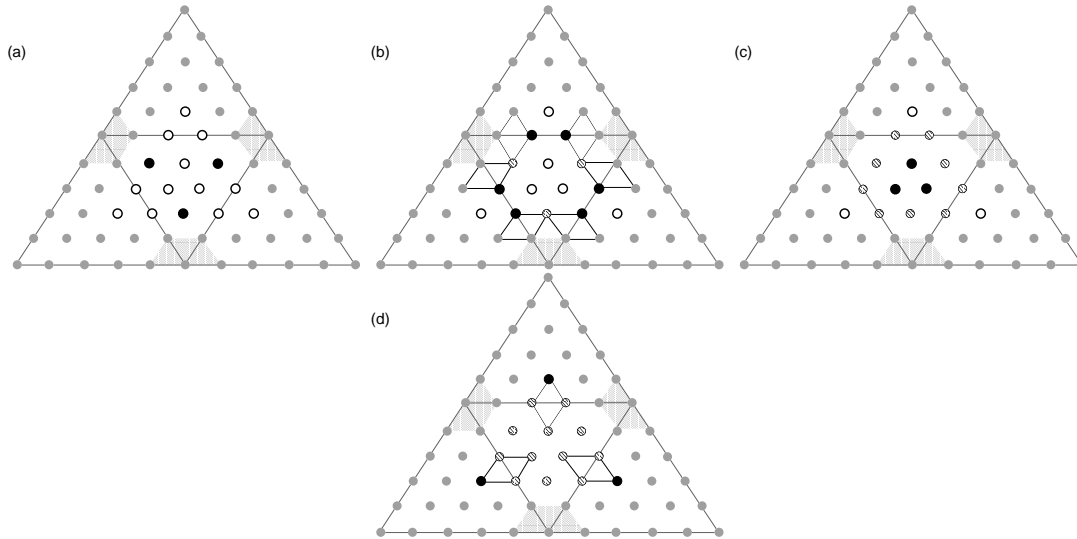


FIG. 3.13 – Quatre étapes pour remplir la macro-facette M avec la continuité C^1 : (a) choisir les trois twists de S^m . (b) calculer le troisième et le quatrième point de Bézier de chaque arête interne en utilisant les conditions C^1 . (c) choisir les trois derniers points de Bézier de S^m . (d) calculer les trois points de Bézier restants, en utilisant les conditions de la continuité C^1 .

Chapitre 4

Première généralisation de la méthode du 4-split

La méthode du chapitre précédent possède plusieurs propriétés importantes. La plupart de ces propriétés proviennent de l'utilisation de facettes de Bézier triangulaires. Nous en donnons ici une liste brève :

- *Invariance affine* : La surface finale est invariante par les transformations affines du réseau surfacique. Cette propriété est importante en pratique. Car les cartes affines, comme la translation, la rotation et l'homothétie sont très souvent utilisées en pratique. Grâce à la propriété de l'invariance affine, il suffit d'appliquer la transformation affine aux points de contrôle de la surface au lieu de l'appliquer à tous les points de la surface.
- *Représentation explicite* : La surface est représentée par des facettes de Bézier quintiques dont les points de Bézier sont explicitement connus. Cette propriété permet de faire des calculs importants relatifs à la surface, comme par exemple l'évaluation des quantités intrinsèques telles que les courbures gaussiennes et moyennes. Cette propriété n'est pas vérifiée par les algorithmes de subdivision qui définissent une surface lisse par un algorithme de subdivision récursif d'un polyèdre de contrôle. Ces algorithmes ne fournissent pas une paramétrisation explicite de la surface.
- *Paramètres de forme* : La méthode permet l'utilisation d'un nombre de paramètres scalaires et de points de contrôle libres pour modifier ou optimiser localement la forme de la surface.
- *Localité* : La méthode est locale dans le sens où un changement d'un sommet \mathbf{v} du réseau surfacique entraîne le changement des macro-facettes autour de ce sommet ainsi que des macro-facettes autour des sommets voisins de \mathbf{v} , mais laisse toutes les autres macro-facettes inchangées.

La localité d'un schéma d'interpolation est une propriété importante pour au moins deux raisons. D'abord du point de vue numérique, un schéma local, à l'inverse d'un schéma glo-

bal, ne nécessite pas de résolution numérique d'un grand système d'équations. Le présent schéma est explicite et ne repose que sur des combinaisons convexes, ce qui implique une bonne stabilité numérique.

Le deuxième avantage de la localité est dans la rapidité avec laquelle les calculs peuvent être effectués. Dans un système graphique 3D, la surface peut être modifiée interactivement, car la mise à jour de la surface se fait en temps réel et seulement un petit nombre de facettes doivent être recalculées après chaque édition de sommet.

Pourtant la méthode de base du 4-split n'est pas strictement locale dans le sens où une édition d'un sommet change la surface au-delà des macro-facettes adjacentes à ce sommet. Ceci est dû au fait que les courbes frontière et les tangentes trans-frontière sont calculées autour d'un sommet à partir des sommets voisins de celui-ci, comme le montre les équations 3.18, 3.20, 3.28. Ceci implique donc bien que le changement d'un sommet modifie non seulement les macro-facettes adjacentes à ce sommet, mais également les macro-facettes adjacentes aux voisins de ce sommet. Une méthode strictement locale ne devrait modifier que les macro-facettes adjacentes à un sommet lors de son édition.

Dans ce chapitre, nous présentons une première généralisation de la méthode du 4-split (chap. 3). Elle a pour but de la rendre strictement locale. La généralisation que nous allons proposer repose sur la remarque suivante : le voisinage d'un sommet \mathbf{v} , intervenant dans tous les calculs autour de ce sommet, peut être sans problème remplacé par un autre ensemble de n points, non nécessairement égaux aux points voisins. La méthode devient alors strictement locale. De plus l'interpolation des vecteurs normaux aux sommets devient possible, ce qui offre un outil supplémentaire de design.

4.1 Le schéma du 4-split n'est pas strictement local

Dans la section précédente, nous avons remarqué que la localité de la méthode du 4-split est due à la construction des courbes frontière et des tangentes trans-frontière. Nous allons ici préciser cette remarque. Soit \mathbf{v} un sommet de degré n du réseau surfacique, \mathbf{v} est donc commun à n faces du réseau surfacique et possède n sommets voisins \mathbf{p}_i . La courbe frontière allant de \mathbf{v} à \mathbf{p}_i est composée de 2 morceaux de cubiques. Les points de Bézier du premier segment paramétré sur $[0, \frac{1}{2}]$ sont $b_0^i, b_1^i, b_2^i, b_3^i$. Le deuxième segment est calculé à partir du sommet \mathbf{p}_i . Les points b_0^i, b_1^i, b_2^i ne sont pas libres. Ils ont été construits sous les conditions G^1 suivant les formules suivantes :

$$\begin{aligned}\bar{\mathbf{b}}^0 &= \alpha \bar{\mathbf{v}} + \mathbf{B}^0 \bar{\mathbf{p}}, \\ \bar{\mathbf{b}}^1 &= \alpha \bar{\mathbf{v}} + \mathbf{B}^1 \bar{\mathbf{p}}, \\ \bar{\mathbf{b}}^2 &= [(\gamma_0 + \gamma_1)\alpha + \frac{\gamma_2}{3}] \bar{\mathbf{v}} + \mathbf{B}^2 \bar{\mathbf{p}},\end{aligned}\tag{4.1}$$

où $\bar{\mathbf{p}} = [\mathbf{p}_1, \dots, \mathbf{p}_n]^T$, $\bar{\mathbf{b}}_0 = [\mathbf{b}_0^1, \dots, \mathbf{b}_0^n]^T$, etc. sont des vecteurs $(n \times 3)$ composés respectivement de n vecteurs \mathbf{p}_i , \mathbf{b}_0^i , etc. Les matrices $\mathbf{B}^0, \mathbf{B}^1, \mathbf{B}^2$ sont données par : (voir sect. 3.7)

$$\begin{aligned} \mathbf{B}_{ij}^0 &= \frac{1-\alpha}{n}, \\ \mathbf{B}_{ij}^1 &= \frac{1}{n} \left[1 - \alpha + \beta \cos\left(\frac{2\pi(j-i)}{n}\right) \right], \quad i, j = 1, \dots, n. \\ \mathbf{B}_{ij}^2 &= \frac{1}{n} \left[(\gamma_0 + \gamma_1)(1 - \alpha) + \gamma_1 \beta \cos\left(\frac{2\pi(j-i)}{n}\right) \right] + \gamma_2 \begin{cases} 1/6 & \text{si } j = i - 1, i + 1 \\ 1/3 & \text{si } j = i \\ 0 & \text{sinon.} \end{cases} \end{aligned} \quad (4.2)$$

La dérivée trans-frontière le long de la courbe frontière $M^i(u_i, 0)$ est donnée par : (voir sect. 3.9)

$$[D_{\mathbf{u}_{i+1}} M^i](u_i, 0) = \Phi_i(u_i) [D_{\mathbf{u}_i} M^i](u_i, 0) + \Psi_i(u_i) \mathbf{V}_i(u_i). \quad (4.3)$$

Rappelons que $\mathbf{V}_i(u_i)$ est une fonction vectorielle composée de deux morceaux quadratiques. Comme pour les courbes frontière, chaque morceau de \mathbf{V}_i est construit à partir du sommet qui lui correspond sous la forme de Bézier suivante :

$$\mathbf{V}_i(u_i) = \sum_{j=0}^2 \mathbf{v}_j^i B_j^2(u_i), \quad u_i \in [0, \frac{1}{2}], \quad i = 1, \dots, n,$$

où

$$\begin{aligned} \bar{\mathbf{v}}_0 &= \mathbf{V}^0 \bar{\mathbf{p}}, \\ \bar{\mathbf{v}}_1 &= \mathbf{V}^1 \bar{\mathbf{p}}, \\ \bar{\mathbf{v}}_2 &= \frac{1}{2} \bar{\mathbf{v}}_1 + \frac{1}{2} \bar{\mathbf{v}}_1^k, \end{aligned} \quad (4.4)$$

où $\bar{\mathbf{v}}_1^k$ est le point de Bézier du deuxième segment de \mathbf{V}_i obtenu à partir de \mathbf{p}_i (k est l'indice de \mathbf{v} dans le voisinage de \mathbf{p}_i) et $\mathbf{V}^0, \mathbf{V}^1$ sont les deux matrices $n \times n$ suivantes :

$$\begin{aligned} \mathbf{V}_{ij}^0 &= \frac{6\beta}{n} \sin \frac{2\pi(j-i)}{n}, \\ \mathbf{V}_{ij}^1 &= \frac{6\beta}{n\Psi_i^0} \left[(\Phi^1 - 8\Phi^0 + 4\gamma_1\Phi^0) \tan \frac{\pi}{n} - \Psi_i^1 \right] \sin \frac{2\pi(j-i)}{n} \\ &\quad + \frac{4}{\Psi_i^0} \gamma_2 \Phi^0 \begin{cases} 1/6 & \text{si } j = i - 1, i + 1 \\ 1/3 & \text{si } j = i \\ 0 & \text{sinon} \end{cases} \quad i, j = 1, \dots, n, \end{aligned} \quad (4.5)$$

avec $\Phi^0 = \cos \frac{2\pi}{n}$, $\Phi^1 = 1 + 2 \cos \frac{2\pi}{n}$, $\Psi^0 = \sin \frac{2\pi}{n}$, $\Psi^1 = \sin \frac{2\pi}{n_i} - \sin \frac{2\pi}{n}$ et n_i le degré de \mathbf{p}_i .

Ainsi, nous pouvons mesurer la localité de la méthode grâce à la dépendance des vecteurs $\bar{\mathbf{b}}_0, \bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \bar{\mathbf{v}}_0$ et $\bar{\mathbf{v}}_1$ des sommets du réseau surfacique. Les formules 4.1 et 4.4

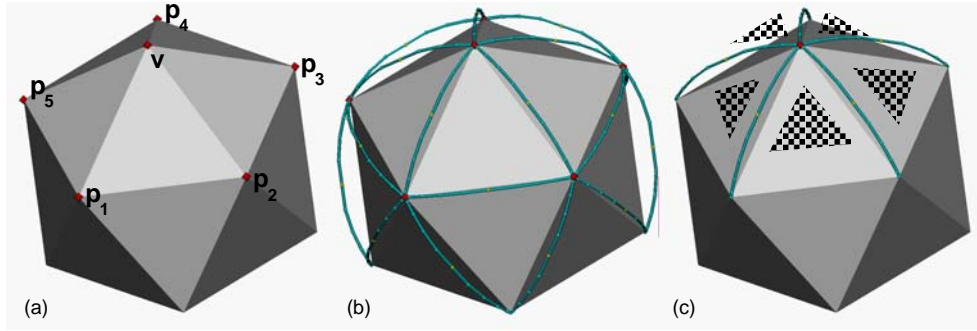


FIG. 4.1 – un polyèdre régulier en entrée. (a) Le voisinage de \mathbf{v} . (b) Localité de l'ancienne méthode : si nous déplaçons le sommet \mathbf{v} , toutes les courbes frontière passant par \mathbf{v} ou par l'un de ses voisins \mathbf{p}_i doivent être recalculées. (c) Localité de la deuxième méthode : n facettes seulement dépendent de \mathbf{v} . L'utilisation du concept de voisinage virtuel a rendu la méthode strictement locale.

montrent clairement la dépendance en fonction de \mathbf{v} et de $\bar{\mathbf{p}}$. Donc toutes les facettes passant par \mathbf{v} ou par l'un des sommets \mathbf{p}_i sont dépendantes, i.e. un changement de \mathbf{v} entraînera le changement de toutes les courbes frontière et les tangentes trans-frontière modifiées par l'édition d'un sommet. La figure 4.1(b) montre les courbes frontière de ces facettes.

4.2 Localisation du schéma du 4-split

Nous voulons généraliser la méthode de subdivision uniforme afin d'obtenir une méthode d'interpolation *strictement locale* en gardant toujours les mêmes étapes de la méthode de base (cf. chap. 3). La notion d'interpolation strictement locale (voir [98]) signifie que chaque macro-facette dépend seulement des données de ces trois sommets, i.e. le changement d'un sommet n'a une influence que sur les facettes qui l'entourent. Fig. 4.1(c) montre les facettes modifiées par un changement de \mathbf{v} dans le cas d'une stricte localité.

En observant les équations 4.1 et 4.4, on remarque des relations linéaires explicites entre les vecteurs $\bar{\mathbf{b}}_0, \bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \bar{\mathbf{v}}_0, \bar{\mathbf{v}}_1$ et $\bar{\mathbf{p}}$. Puisque nous voulons conserver les mêmes étapes que lors de la méthode du 4-split, la seule manière de supprimer la dépendance entre ces vecteurs et $\bar{\mathbf{p}}$ est de remplacer $\bar{\mathbf{p}}$ par un autre vecteur $\bar{\mathbf{p}}^*$ de $\mathbb{R}^{n \times 3}$. Ce vecteur sera appelé un *voisinage virtuel* de \mathbf{v} . C'est un voisinage virtuel dans le sens où il vient remplacer le voisinage effectif, uniquement pour le calcul des courbes frontière et des dérivées trans-frontière. La question qui se pose est alors de voir si le changement du vecteur $\bar{\mathbf{p}}$ par le vecteur $\bar{\mathbf{p}}^*$ affecte ou non la continuité G^1 de la surface et le fait que la surface interpole

le réseau donné en entrée. Nous allons voir ici que le changement du voisinage réel par un voisinage virtuel, pour le calcul des courbes frontière et des dérivées trans-frontière, n'affecte ni l'interpolation ni les conditions G^1 entre les facettes. Il faut savoir que les seules conditions portant sur les points $\bar{\mathbf{b}}_0, \bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2$, et sur les vecteurs $\bar{\mathbf{v}}_0, \bar{\mathbf{v}}_1$ sont les conditions de continuité G^1 autour de \mathbf{v} . Dans ce qui suit, nous allons démontrer que ces conditions restent vérifiées après l'introduction du voisinage virtuel à la place du voisinage réel.

Lemme 2 *Les vecteurs*

$$\begin{cases} \bar{\mathbf{b}}_0 &= \alpha \mathbf{v} + \mathbf{B}^0 \mathbf{p}^*, \\ \bar{\mathbf{b}}_1 &= \alpha \bar{\mathbf{v}} + \mathbf{B}^1 \bar{\mathbf{p}}^*, \\ \bar{\mathbf{b}}_2 &= [(\gamma_0 + \gamma_1)\alpha + \frac{\gamma_2}{3}] \bar{\mathbf{v}} + \mathbf{B}^2 \bar{\mathbf{p}}^* \end{cases} \quad (4.6)$$

sont dans l'espace image de la matrice twist \mathbf{M}_t du système 3.11.

Preuve:

Il suffit de démontrer qu'il existe trois vecteurs $\tilde{\mathbf{b}}_i$ de $\mathbb{R}^{n \times 3}$ tels que $\mathbf{M}_t \tilde{\mathbf{b}}_i = \bar{\mathbf{b}}_i$ pour $i = 0, 1, 2$. Voyons d'abord la caractérisation de l'espace image de \mathbf{M}_t (voir aussi [84] et [65]).

Un vecteur $\bar{\mathbf{b}}$ appartient à $\text{Im}(\mathbf{M}_t)$ si et seulement s'il existe un vecteur $\tilde{\mathbf{b}} \in \mathbb{R}^{n \times 3}$ tel que $\mathbf{b}^i = \frac{1}{2} \tilde{\mathbf{b}}^{i-1} + \frac{1}{2} \tilde{\mathbf{b}}^i$, pour $i \in \{1, \dots, n\}$.

Pour le vecteur $\bar{\mathbf{b}}_0$, il suffit de prendre $\tilde{\mathbf{b}}_0 = \bar{\mathbf{b}}_0$, car $\frac{1}{2} \mathbf{b}_{i-1}^0 + \frac{1}{2} \mathbf{b}_i^0 = \mathbf{b}_i^0$.

Afin de trouver $\tilde{\mathbf{b}}_1$, nous devons résoudre le système $\frac{1}{2} \tilde{\mathbf{b}}_1^{i-1} + \frac{1}{2} \tilde{\mathbf{b}}_1^i = \mathbf{b}_1^i$, $i = 1, \dots, n$ en $\tilde{\mathbf{b}}_1$. Une solution peut être trouvée dans [84], page 315, en remplaçant \mathbf{p} par \mathbf{p}^* :

$$\tilde{\mathbf{b}}_1^i = \alpha \bar{\mathbf{v}} + \frac{1}{n} \sum_{j=1}^n \left[(1 - \alpha) + \beta \left(\cos \frac{2\pi(j-i)}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi(j-i)}{n} \right) \right] \mathbf{p}_j^*.$$

Pour trouver $\tilde{\mathbf{b}}_2$, rappelons que $\bar{\mathbf{b}}_2$ s'écrit $\gamma_0 \bar{\mathbf{b}}_0 + \gamma_1 \bar{\mathbf{b}}_1 + \gamma_2 \bar{\mathbf{d}}$ avec $\mathbf{d}^i = \frac{1}{6}(2\mathbf{v} + \mathbf{p}_{i-1}^* + 2\mathbf{p}_{i-1}^* + \mathbf{p}_{i+1}^*)$. Mais d'après Loop [84], $\bar{\mathbf{d}}$ appartient à l'espace image de \mathbf{M}_t et le vecteur $\tilde{\mathbf{d}} = (\frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{p}_i^* + \frac{1}{3}\mathbf{p}_{i+1}^*)_{i=1, \dots, n}$ vérifie $\mathbf{M}_t \tilde{\mathbf{d}} = \bar{\mathbf{d}}$. Nous pouvons donc écrire

$$\begin{aligned} \mathbf{M}_t(\gamma_0 \tilde{\mathbf{b}}_0 + \gamma_1 \tilde{\mathbf{b}}_1 + \gamma_2 \tilde{\mathbf{d}}) &= \gamma_0 \mathbf{M}_t \tilde{\mathbf{b}}_0 + \gamma_1 \mathbf{M}_t \tilde{\mathbf{b}}_1 + \gamma_2 \mathbf{M}_t \tilde{\mathbf{d}} \\ &= \gamma_0 \bar{\mathbf{b}}_0 + \gamma_1 \bar{\mathbf{b}}_1 + \gamma_2 \bar{\mathbf{d}} \\ &= \bar{\mathbf{b}}_2. \end{aligned}$$

On peut alors prendre $\tilde{\mathbf{b}}_2 = \gamma_0 \tilde{\mathbf{b}}_0 + \gamma_1 \tilde{\mathbf{b}}_1 + \gamma_2 \tilde{\mathbf{d}}$. \square

Lemme 3 *Les vecteurs twists compatibles sont donnés explicitement par*

$$\begin{aligned} \mathbf{t}_i &= 8\gamma_2 \Phi^0(1-3\alpha)\mathbf{v} + 8\gamma_2 \Phi^0(\mathbf{p}_i^* + \mathbf{p}_{i+1}^*) \\ &+ \frac{1}{n} \sum_{j=1}^n \left[-24\gamma_2 \Phi^0(1-\alpha) + \beta(6\Phi^1 + (24\gamma_1 - 48)\Phi^0) \left(\cos \frac{2\pi(j-i)}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi(j-i)}{n} \right) \right] \mathbf{p}_j^*, \\ & \qquad \qquad \qquad i = 1, \dots, n. \end{aligned} \quad (4.7)$$

Preuve:

Cette formule peut être trouvée d'une manière analogue à celle de la section 3.8 en remplaçant $\bar{\mathbf{p}}$ par $\bar{\mathbf{p}}^*$ dans 3.21. \square

Lemme 4 *La dérivée trans-frontière donnée par 4.3 en remplaçant $\bar{\mathbf{p}}$ par $\bar{\mathbf{p}}^*$ satisfait les conditions du raccordement G^1 (données par 3.1 et 3.7), interpole les valeurs des twists aux points extrémités, et est compatible avec le réseau des courbes frontière.*

Preuve:

La démonstration de ce lemme suit les mêmes étapes que dans la section 3.9 mais en remplaçant $\bar{\mathbf{p}}$ par $\bar{\mathbf{p}}^*$. \square

Donc le remplacement de $\bar{\mathbf{p}}$ par $\bar{\mathbf{p}}^*$ engendre toujours une solution de notre problème d'interpolation et permet cette fois un schéma strictement local. Les courbes frontière et les tangentes trans-frontière sont définies par leurs points de Bézier suivant

$$\left\{ \begin{array}{l} \bar{\mathbf{b}}^0 = \alpha \mathbf{v} + \mathbf{B}^0 \bar{\mathbf{p}}^*, \\ \bar{\mathbf{b}}^1 = \alpha \bar{\mathbf{v}} + \mathbf{B}^1 \bar{\mathbf{p}}^*, \\ \bar{\mathbf{b}}^2 = [(\gamma_0 + \gamma_1)\alpha + \frac{\gamma_2}{3}] \bar{\mathbf{v}} + \mathbf{B}^2 \bar{\mathbf{p}}^*, \\ \bar{\mathbf{b}}_3 = \frac{1}{2} \bar{\mathbf{b}}_2 + \frac{1}{2} \bar{\mathbf{b}}_2^k, \end{array} \right. \quad \left\{ \begin{array}{l} \bar{\mathbf{v}}_0 = \mathbf{V}^0 \bar{\mathbf{p}}^*, \\ \bar{\mathbf{v}}_1 = \mathbf{V}^1 \bar{\mathbf{p}}^*, \\ \bar{\mathbf{v}}_2 = \frac{1}{2} \bar{\mathbf{v}}_1 + \frac{1}{2} \bar{\mathbf{v}}_1^k, \end{array} \right. \quad (4.8)$$

où $\mathbf{B}^0, \mathbf{B}^1, \mathbf{B}^2, \mathbf{V}^0, \mathbf{V}^1$ sont les matrices déjà données en 4.2 et 4.5.

Du point de vue algorithmique, la surface interpolante est construite par les trois étapes principales habituelles :

1. construction du réseau de courbes, G^1 compatible aux sommets qu'il interpole,
2. construction des rubans de tangentes trans-frontière, compatibles avec le réseau de courbes et avec la continuité G^1 ,
3. calcul des points de Bézier de la frontière des facettes ainsi que de la première rangée intérieure en élevant le degré des courbes frontière et des dérivées trans-frontière,
4. remplissage de chaque facette avec une continuité C^1 entre ses quatre sous-facettes.

Seules les étapes (1) et (2) sont affectées par l'introduction du voisinage virtuel. Pour chaque sommet de degré n , on construit les n morceaux de courbes et les n morceaux des dérivées trans-frontière. Une traversée de toutes les arêtes fait ensuite les jonctions entre ces morceaux en déterminant les points milieux $\bar{\mathbf{b}}_3$ et $\bar{\mathbf{v}}_2$. Enfin l'étape trois est effectuée.

L'utilisation du voisinage virtuel pour chaque sommet entraîne une modification dans la structure de donnée par la création d'un nouveau tableau pour le stockage de $\bar{\mathbf{p}}^*$. En particulier, il n'est plus nécessaire, au moment des calculs à effectuer en chacun des sommets, d'utiliser une routine traversant le voisinage de ce sommet: un simple parcours séquentiel du nouveau tableau stockant le voisinage virtuel suffit. Les étapes (3) et (4) sont identiques à celles du chap. 3. Nous obtenons ainsi un interpolant strictement local car les étapes (1) et (2) n'utilisent que le sommet lui-même du réseau surfacique, et pas ses sommets voisins.

L'utilisation d'un voisinage virtuel introduit d'autres propriétés importantes, en plus de la stricte localité. Ces propriétés sont principalement liées au nombre de degrés de liberté. Ceux-ci ont augmenté, car le voisinage virtuel rajoute en chaque sommet n points libres en plus des trois paramètres scalaires $\beta, \gamma_1, \gamma_2$. Il est important de savoir utiliser ces nouveaux degrés de liberté pour contrôler la forme, ou pour créer localement des outils de modélisation. Ce sera le sujet des sections suivantes et du chapitre 6.

4.3 Degrés de liberté de la méthode

En remplaçant les points du voisinage réel de $\mathbf{v}, \mathbf{p}_1, \dots, \mathbf{p}_n$, par les points du voisinage virtuel $\mathbf{p}_1^*, \dots, \mathbf{p}_n^*$, nous avons ajouté de nouveaux degrés de liberté en chaque sommet. Le choix de ces degrés de liberté est essentiel pour contrôler la forme de la surface finale. Nous pouvons utiliser les points \mathbf{p}_i^* pour interpoler des vecteurs normaux aux sommets du réseau surfacique. Nous pouvons aussi les utiliser avec les paramètres $\beta, \gamma_1, \gamma_2$ pour réaliser des procédures de lissage.

4.3.1 Choix optimal des points \mathbf{b}_1^i

La deuxième équation de 4.8 exprime la relation entre le vecteur des \mathbf{b}_1^i et les points libres du voisinage virtuel \mathbf{p}_i^* . Le nombre effectif des degrés de liberté dans le choix des \mathbf{b}_1^i dépend donc du rang de la matrice \mathbf{B}^1 , qui vaut 2, comme nous allons le montrer dans la suite. Nous aurons donc deux degrés de liberté vectoriels dans le choix des \mathbf{b}_1^i . Nous montrerons dans la section suivante que ces deux vecteurs de liberté permettent d'interpoler un vecteur normal en \mathbf{v} .

4.3.2 Interpolation des vecteurs normaux

Nous savons que la formule $\bar{\mathbf{b}}^1 = \alpha \bar{\mathbf{v}} + \mathbf{B}^1 \bar{\mathbf{p}}^*$ implique que les points \mathbf{b}_i^1 appartiennent au même plan tangent passant par \mathbf{b}_i^0 . Nous nous intéressons maintenant à la définition de ce plan tangent et à sa relation avec les points \mathbf{p}_i^* .

Lemme 5 \mathbf{B}^1 est une matrice symétrique et de rang égal à 2.

Preuve:

Nous avons

$$\mathbf{B}_{ij}^1 = \frac{\beta}{n} \cos\left(2 \frac{\pi(j-i)}{n}\right) = \frac{\beta}{n} \cos\left(2 \frac{\pi(i-j)}{n}\right) = \mathbf{B}_{ji}^1.$$

\mathbf{B}^1 est donc symétrique. Démontrons que le rang de \mathbf{B}^1 est égal à 2. Pour cela, nous allons démontrer que chaque vecteur ligne de \mathbf{B}^1 s'écrit sous forme d'une combinaison linéaire de deux vecteurs $\mathbf{V}_1, \mathbf{V}_2$ de \mathbb{R}^n . En effet

$$\mathbf{B}_{ij}^1 = \frac{\beta}{n} \cos\left(2 \frac{\pi(j-1)}{n}\right) \cos\left(2 \frac{\pi(i-1)}{n}\right) + \frac{\beta}{n} \sin\left(2 \frac{\pi(j-1)}{n}\right) \sin\left(2 \frac{\pi(i-1)}{n}\right). \quad (4.9)$$

Posons

$$\mathbf{V}_1 = \begin{bmatrix} \cos\left(2 \frac{(0)\pi}{n}\right) \\ \vdots \\ \cos\left(2 \frac{(n-1)\pi}{n}\right) \end{bmatrix}, \quad \mathbf{V}_2 = \begin{bmatrix} \sin\left(2 \frac{(0)\pi}{n}\right) \\ \vdots \\ \sin\left(2 \frac{(n-1)\pi}{n}\right) \end{bmatrix}.$$

Il est clair que $\mathbf{V}_1, \mathbf{V}_2$ sont linéairement indépendants. On en déduit que la $i^{\text{ème}}$ ligne de \mathbf{B}^1 s'écrit $\mathbf{B}_i^1 = \frac{\beta}{n} \cos\left(2 \frac{\pi i}{n}\right) \mathbf{V}_1 + \frac{\beta}{n} \sin\left(2 \frac{\pi i}{n}\right) \mathbf{V}_2$. Autrement dit, les colonnes de \mathbf{B}^1 engendrent un sous espace vectoriel de \mathbb{R}^n de base $\{\mathbf{V}_1, \mathbf{V}_2\}$. Il est donc de dimension 2, qui est par définition le rang de \mathbf{B}^1 .

Corollaire 2 Le plan tangent en un sommet \mathbf{v} est engendré par les deux vecteurs suivants :

$$\mathbf{a} = \sum_{j=1}^n \cos \frac{2\pi(j-1)}{n} \mathbf{p}_j^*, \quad \mathbf{b} = \sum_{j=1}^n \sin \frac{2\pi(j-1)}{n} \mathbf{p}_j^*. \quad (4.10)$$

En effet, les vecteurs tangents autour de \mathbf{v} sont donnés par :

$$\mathbf{r}_i^1 = 6(\mathbf{b}_1^i - \mathbf{v}), \quad (4.11)$$

(le terme 6 provient du fait qu'on dérive sur l'intervalle $[0, \frac{1}{2}]$).

De 4.9, on en déduit que

$$\begin{aligned} \mathbf{r}_i^1 &= \frac{6\beta}{n} \left(c_i \sum_{j=1}^n \cos\left(2 \frac{\pi(j-1)}{n}\right) \mathbf{p}_j^* + s_i \sum_{j=1}^n \frac{\beta}{n} \sin\left(2 \frac{\pi(j-1)}{n}\right) \mathbf{p}_j^* \right) \\ &= \frac{6\beta}{n} (c_i \mathbf{a} + s_i \mathbf{b}), \end{aligned} \quad (4.12)$$

avec $c_i = \cos(2 \frac{\pi(i-1)}{n})$ et $s_i = \sin(2 \frac{\pi(i-1)}{n})$.

Remarque 9 Les vecteurs tangents \mathbf{r}_i^1 sont donnés par (en remplaçant \mathbf{b}_1^i dans 4.11) :

$$\mathbf{r}_i^1 = \sum_{j=1}^n \frac{\beta}{n} \cos(2 \frac{\pi(j-i)}{n}) \mathbf{p}_j^*.$$

Cette formule peut aussi s'écrire sous la forme suivante (voir sect. 3.7.2) :

$$\mathbf{r}_i^1 = \sum_{j=1}^n \frac{\beta}{n} \cos(2 \frac{\pi(j-1)}{n}) \mathbf{p}_{i+j-1}^*. \quad (4.13)$$

On remarque que cette dernière formule attribue à \mathbf{p}_i^* le facteur $\frac{\beta}{n}$ qui est supérieur à tous les autres facteurs des \mathbf{p}_j^* , $j \neq i$. Ceci implique que \mathbf{r}_i^1 est plus proche de l'arête virtuelle $[\mathbf{v}\mathbf{p}_i^*]$ que de toutes les autres arêtes $[\mathbf{v}\mathbf{p}_j^*]$, $j \neq i$. Ce qui explique l'intention de construire des courbes frontière proches des arêtes qui leur sont associées. On remarque aussi la division par n qui est naturelle, car si le nombre de sommets dans 4.13 est assez élevé alors la longueur de \mathbf{r}_i^1 est normalisée par cette division. Le paramètre β est d'ailleurs introduit pour contrôler les longueurs de ces vecteurs.

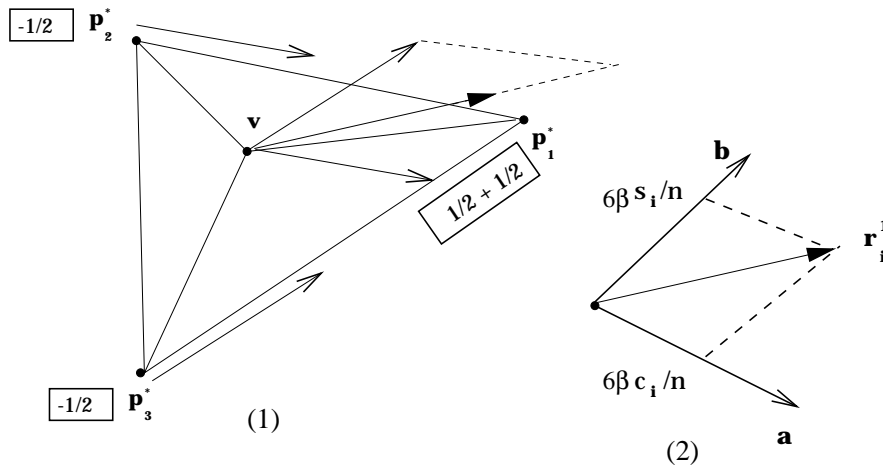


FIG. 4.2 – Deux manières pour construire \mathbf{r}_i^1 . (1.) Utilisation de la formule 4.13 dans laquelle chaque point virtuel \mathbf{p}_{i+j-1}^* est pondéré par le coefficient $(6\beta/n) \cos(2 \frac{\pi(j-1)}{n})$. (2.) Utilisation de la base $\{\mathbf{a}, \mathbf{b}\}$ du plan tangent donnée par 4.10: \mathbf{r}_i^1 est construit avec les coordonnées $6\beta/n c_i$ et $6\beta/n s_i$

Fig. 4.2 montre la construction des vecteurs tangents \mathbf{r}_i^1 pour $n = 3$ de deux manières équivalentes basées respectivement sur les deux formules 4.13 et 4.12.

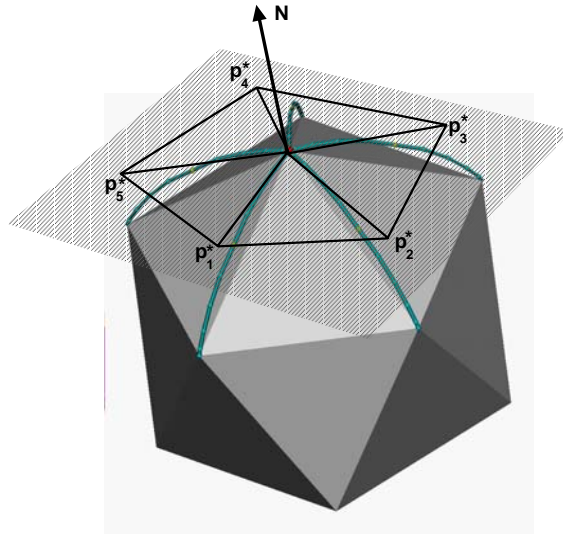


FIG. 4.3 – *Interpolation du vecteur normal: les points du voisinage virtuel \mathbf{p}_i^* appartiennent au plan passant par \mathbf{v} et orthogonal à \mathbf{N} .*

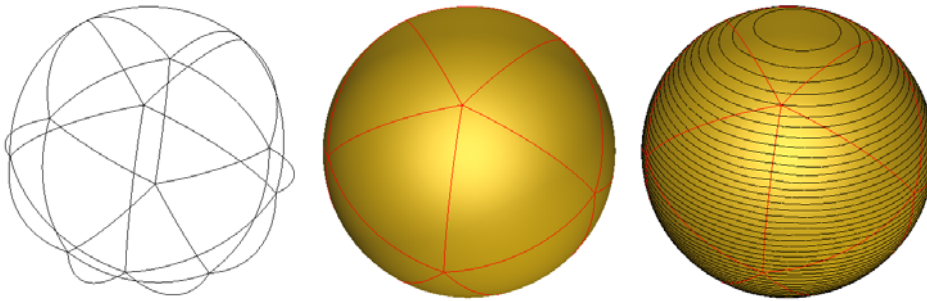


FIG. 4.4 – *Interpolation de l'icosaèdre avec le voisinage virtuel.*

Si le plan tangent en \mathbf{v} est imposé par la donnée d'un vecteur normal \mathbf{N} , alors nous pouvons choisir tous les points \mathbf{p}_i^* dans ce plan. De cette manière, tous les vecteurs \mathbf{r}_i^1 sont dans ce plan et l'interpolation position-normale est assurée. Reste à définir la manière suivant laquelle les points \mathbf{p}_i^* sont distribués dans ce plan autour de \mathbf{v} . Ce problème sera traité dans le chapitre 6, sect. 6.2.5.

Les résultats présentés dans ce chapitre ont fait l'objet d'une première publication dans [67].

4.4 Quelques résultats

Nous considérons d'abord l'exemple d'un icosaèdre. Les points de contrôle des courbes frontière et des tangentes trans-frontière sont calculés d'après les équations 4.8. Les paramètres scalaires ont les valeurs suivantes¹ : $\alpha = 1$, $\beta = 0.12$, $\gamma_0 = -3.7$, $\gamma_1 = 4.6$, $\gamma_2 = 0.1$. Ils sont choisis identiques pour tous les sommets à cause des symétries de l'icosaèdre. Le voisinage virtuel a été choisi pour que les normales de la sphère unité soient interpolées. La figure 4.4 montre une bonne approximation de la sphère unité avec une erreur L_∞ entre la surface finale et la sphère unité égale à 0.0033. Cette erreur est calculée suivant la formule

$$\max_{(x,y,z) \in D} |x^2 + y^2 + z^2 - 1|,$$

où D est l'ensemble des points évalués de la surface. Les lignes isophotes, à droite, montrent la continuité G^1 de la surface.

Le deuxième exemple, fig. 4.5, montre la stricte localité de la méthode. En déplaçant l'un des sommets du réseau précédent, nous remarquons que seules les macro-facettes incidentes à ce sommet ont été modifiées par ce déplacement. Notons que les courbes opposées à ce sommet et les tangentes trans-frontière correspondantes ne sont pas affectées par ce déplacement.

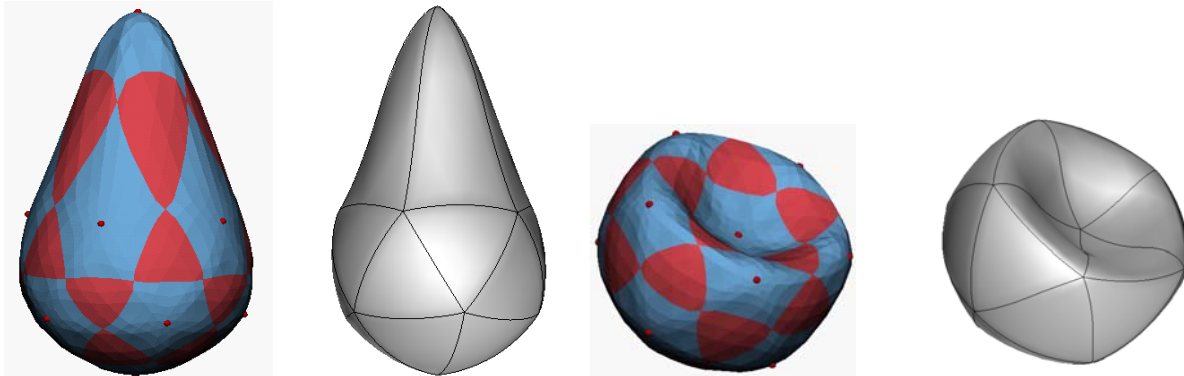


FIG. 4.5 – la deuxième méthode est strictement locale.

Le dernier exemple, fig. 4.6, montre l'utilisation du concept de voisinage virtuel pour l'interpolation des vecteurs normaux aux sommets d'un réseau surfacique de topologie arbitraire. L'interpolation position-normale est illustrée avec les vecteurs normaux et les isophotes.

1. Ces valeurs ont été trouvées après plusieurs tests numériques

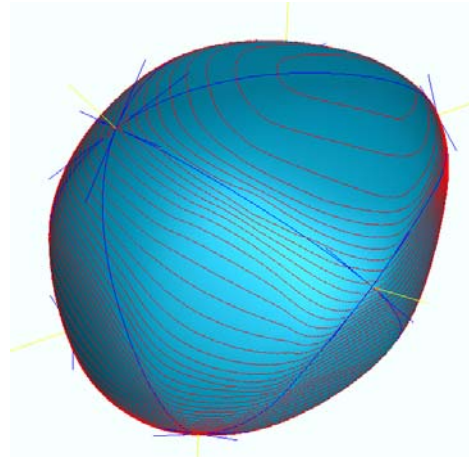


FIG. 4.6 – *Interpolation des vecteurs normaux aux sommets.*

4.5 Paramètres de forme

Si l'interpolation des vecteurs normaux n'est pas imposée, alors les points \mathbf{p}_i^* peuvent être utilisés comme paramètres de forme. De même que les paramètres scalaires $\beta, \gamma_1, \gamma_2$, ils influencent la forme des courbes frontière de toutes les facettes.

Il est important de les choisir de manière à obtenir des courbes de forme satisfaisante. Cela est d'ailleurs la préoccupation principale de toutes les contributions dans cette thèse. Bien que la méthode du 4-split de base offre un nombre important de paramètres libres, les effets de design sont néanmoins limités. Pour le design des courbes frontière uniquement trois paramètres scalaires sont disponibles pour n morceaux de courbe autour de chaque sommet. La première généralisation qui introduit le voisinage virtuel pour chaque sommet est un premier pas vers un meilleur contrôle des courbes frontière.

Il est intéressant de noter qu'une méthode d'interpolation strictement locale nécessite tout particulièrement un réseau de courbes frontière bien conçu pour que la surface finale ait une forme satisfaisante.

Le chapitre 6 sera d'ailleurs entièrement dédié à la question du choix de ces paramètres de forme ou des degrés de liberté afin d'optimiser la forme de la surface. Cette optimisation consiste soit en la minimisation d'une fonctionnelle d'énergie quadratique, soit en l'utilisation de règles heuristiques basées sur la géométrie du réseau surfacique.

Chapitre 5

Deuxième Généralisation de la méthode 4-split

5.1 Introduction

La méthode du 4-split, détaillée dans le chapitre 3, a permis l'interpolation locale des sommets d'un réseau surfacique de topologie arbitraire avec des facettes de Bézier quintiques. La stratégie de la méthode est de construire autour de chaque sommet et pour chaque face quatre facettes quintiques en subdivisant, d'une manière uniforme, le triangle domaine. Cette subdivision respecte la forme des faces triangulaires et ne subdivise pas les angles des macro-facettes. La subdivision de type Clough-Tocher, à l'inverse, allonge les sous-facettes deux fois plus que les faces de départ. Ceci peut résulter en des oscillations indésirables sur la surface finale. De plus cette subdivision uniforme a permis l'introduction de plusieurs paramètres de forme donnant plus de souplesse pour le contrôle local de la forme. L'introduction de la notion de voisinage virtuel, dans le chapitre 4, a considérablement augmenté le nombre de paramètres de forme, donnant naissance à une nouvelle méthode strictement locale, permettant de satisfaire plusieurs contraintes de forme. Elle peut être utilisée interactivement afin de modeler localement la surface et satisfaire directement les propriétés géométriques voulues, ou alors elle peut être exécutée automatiquement avec des méthodes heuristiques robustes (expliquées dans la section 6.2) effectuant le choix optimal de ces paramètres de forme. Cette méthode est à considérer comme une version localisée de la méthode du 4-split. Elle permet aussi l'interpolation des plans tangents aux sommets du réseau surfacique. Dans ce chapitre, nous allons proposer une deuxième généralisation de la méthode du 4-split, incluant également la méthode précédente.

Malgré la simplicité de son schéma d'interpolation et malgré ses résultats satisfaisants, la méthode du chap. 4 n'exploite pas pleinement la structure résultante de la méthode du

4-split. Si les paramètres α , β , γ_1 et γ_2 ont en majorité une interprétation géométrique directe (voir la fin de la section 3.8), il n'en est pas de même avec le voisinage virtuel. En effet autour d'un sommet de degré n , nous n'avons pas exactement n points libres dans le choix des vecteurs tangents des courbes frontière partant de ce sommet. De même pour les degrés de liberté dans le choix des dérivées secondes, le nombre effectif des degrés de liberté dépend du rang des matrices \mathbf{B}^1 et \mathbf{B}^2 de la section 4.3.

En s'appuyant toujours sur la même stratégie de la méthode de subdivision uniforme du chapitre 3, nous allons mieux l'exploiter et nous allons étudier de manière théorique et générale le problème de compatibilité de twist. La solution de twist proposée par la méthode de subdivision uniforme de base ou par celle de Loop [84] deviennent des cas particuliers de celle proposée dans ce chapitre. Les paramètres de forme de cette nouvelle méthode sont géométriquement plus significatifs, permettant ainsi de satisfaire plus facilement les contraintes de forme.

Un grand nombre de ces paramètres accompagnent la construction des courbes frontière. Ceci est important car la méthode est strictement locale et chaque patch de la surface dépend beaucoup de ses trois courbes frontière. Donc la forme de ces courbes influence localement la forme de la surface. On peut donc obtenir une surface localement aplatie si les courbes frontière sont presque plates. Mais si elles sont courbées, la facette correspondante sera aussi courbée. De plus, nous pouvons espérer qu'un meilleur choix du réseau des courbes frontière, débarrassera la surface des problèmes d'oscillations et d'auto-intersections. Les paramètres offerts par la présente méthode sont suffisants pour créer un réseau de courbes frontière bien adapté. On peut contrôler leur forme interactivement, à l'aide d'outils de modélisation permettant de modifier le choix des dérivées premières et secondes aux sommets. La section 6.1 décrit les procédures de modélisation interactive que l'on peut faire pour obtenir localement une forme satisfaisante.

On peut aussi contrôler la forme des courbes frontière à l'aide de quelques méthodes heuristiques utilisant des critères d'optimisation basés sur la minimisation de l'énergie de tension. Cette deuxième généralisation permet aussi de choisir les courbes frontière suivant une autre famille de méthodes heuristiques basées sur les notions de points cibles et de tangentes cibles qui seront expliquées dans la section 5.4.5 et dans les sections 6.2.6 et 6.3.1.

Les résultats présentés dans ce chapitre ont été publiés dans les actes des Journées Modélisation Géométrique, Dijon 2001 [68].

5.2 Relecture du problème de twist

La méthode du chapitre précédent prend en entrée un réseau polyédral de topologie quelconque et le remplace par une surface G^1 continue composée de facettes quintiques.

Le problème de twist est résolu par la construction de courbes frontière satisfaisant les conditions G^1 . En s'appuyant sur ces conditions, nous allons construire, d'une manière plus souple, le réseau des courbes frontière autour de chaque sommet.

En effet, soit \mathbf{v} un sommet quelconque et soit n le nombre de faces qui l'entourent. Les conditions de la continuité G^1 autour de \mathbf{v} sont données, dans la section 3.6, par les conditions 3.6 et 3.7, s'écrivant matriciellement sous la forme :

$$\mathbf{M}_r \bar{\mathbf{r}}^1 = \mathbf{0}, \quad (5.1)$$

$$\mathbf{M}_t \bar{\mathbf{t}} = \Phi^1 \bar{\mathbf{r}}^1 + \Phi^0 \bar{\mathbf{r}}^2, \quad (5.2)$$

avec

$$\mathbf{M}_r = \begin{bmatrix} \phi^0 & -\frac{1}{2} & & & & & -\frac{1}{2} \\ -\frac{1}{2} & \phi^0 & -\frac{1}{2} & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & -\frac{1}{2} & \phi^0 & -\frac{1}{2} \\ -\frac{1}{2} & & & & -\frac{1}{2} & \phi^0 & \end{bmatrix} \quad \text{et} \quad \mathbf{M}_t = \begin{bmatrix} \frac{1}{2} & & & & & & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & & & & & \\ & \cdot & \cdot & & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \\ & & & & & \frac{1}{2} & \frac{1}{2} \\ & & & & & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

On rappelle que $\bar{\mathbf{r}}^1, \bar{\mathbf{r}}^2$ sont les deux vecteurs $(n \times 3)$ composés respectivement des dérivées premières et secondes en \mathbf{v} des courbes frontière autour de \mathbf{v} et $\bar{\mathbf{t}}$ est le vecteur $(n \times 3)$ composé des n vecteurs twists autour de \mathbf{v} .

Pour que le système de twist 5.2 possède toujours une solution en $\bar{\mathbf{t}}$, les vecteurs $\bar{\mathbf{r}}^1$ et $\bar{\mathbf{r}}^2$ doivent appartenir à l'espace image de la matrice \mathbf{M}_t que l'on notera $\text{Im}(\mathbf{M}_t)$, et $\bar{\mathbf{r}}^1$ doit en plus appartenir au noyau de la matrice \mathbf{M}_r que l'on notera $\text{ker}(\mathbf{M}_r)$. La méthode de Loop (cf. sect. 2.9.1) propose un choix particulier de $\bar{\mathbf{r}}^1$ et $\bar{\mathbf{r}}^2$ dans $\text{Im}(\mathbf{M}_t) \cap \text{ker}(\mathbf{M}_t)$ et $\text{Im}(\mathbf{M}_t)$ respectivement. La méthode du chapitre 3 a permis de relâcher les contraintes sur la construction de ces deux vecteurs avec l'utilisation de la subdivision du triangle domaine, mais elle a aussi fait un choix particulier de ces deux vecteurs dans leurs espaces respectifs. Donc afin de construire $\bar{\mathbf{r}}^1$ et $\bar{\mathbf{r}}^2$ avec plus de libertés, il est nécessaire de caractériser les deux espaces $\text{Im}(\mathbf{M}_t)$ et $\text{ker}(\mathbf{M}_r)$. Cela sera facilité par le fait que les deux matrices \mathbf{M}_r et \mathbf{M}_t sont deux matrices circulantes. C'est pour cela que nous commençons d'abord par rappeler la définition et quelques propriétés des matrices circulantes.

5.3 Propriétés des matrices circulantes

(Les propriétés suivantes des matrices circulantes ont été prises de la page web : <http://www.mai.liu.se/evlun/pub/lic/node9.html>)

Une matrice carré de dimension $n \times n$ est dite circulante si elle est de la forme

$$\mathbf{C} = \begin{bmatrix} c_1 & c_n & c_{n-1} & \dots & c_2 \\ c_2 & c_1 & c_n & \dots & c_3 \\ c_3 & c_2 & c_1 & \dots & c_4 \\ \vdots & \vdots & \vdots & & \vdots \\ c_n & c_{n-1} & c_{n-2} & \dots & c_1 \end{bmatrix}.$$

On remarque que \mathbf{C} peut être obtenue de la manière suivante : on laisse fixer la première colonne. Chacune des autres colonnes est obtenue en décalant de manière cyclique chaque élément d'une unité vers le bas. Donc, pour connaître la matrice circulante, il suffit de connaître une de ses colonnes. Plus précisément, nous avons la propriété suivante :

Propriété 1 *La matrice circulante \mathbf{C} peut s'écrire*

$$\mathbf{C} = (\mathbf{c} \quad \mathbf{R}\mathbf{c} \quad \mathbf{R}^2\mathbf{c} \quad \dots \quad \mathbf{R}^{n-1}\mathbf{c}) = c_1 \mathbf{I} + c_2 \mathbf{R} + \dots + c_n \mathbf{R}^{n-1},$$

où

$$\mathbf{c} = (c_1 \quad c_2 \quad \dots \quad c_n)^T, \quad \mathbf{R} = (\mathbf{e}_2 \quad \mathbf{e}_3 \quad \dots \quad \mathbf{e}_n \quad \mathbf{e}_1),$$

et \mathbf{e}_k la $k^{\text{ème}}$ colonne de la matrice identité.

Preuve:

On a $\mathbf{c} = \sum_1^n c_i \mathbf{e}_i$, d'où

$$\mathbf{R}\mathbf{c} = \mathbf{R} \sum_1^n c_i \mathbf{e}_i = \sum_1^n c_i (\mathbf{R} \mathbf{e}_i)$$

mais $\mathbf{R} \mathbf{e}_i = \mathbf{e}_{i+1}$ (modulo n), donc

$$\mathbf{R} \mathbf{c} = \sum_1^n c_i \mathbf{e}_{(i+1)[n]} = (c_n \quad c_1 \quad \dots \quad c_{n-1})^T.$$

La deuxième colonne est donc obtenue en multipliant \mathbf{R} par la première colonne. Chaque colonne de \mathbf{C} est donc obtenue en multipliant sa précédente par \mathbf{R} . \square

Propriété 2 Les vecteurs propres de \mathbf{C} sont donnés par les colonnes de la matrice unitaire ($\mathbf{F}^{-1} = \mathbf{F}^*$) $\mathbf{F} = \frac{1}{\sqrt{n}} (\omega^{kl})_{k,l=0,\dots,n-1}$ où $\omega = \exp(\frac{-2i\pi}{n})$.

Les valeurs propres associées sont données par la matrice diagonale Ω suivante :

$$\Omega = \mathbf{FCF}^{-1} = \sum_{k=1}^n c_k \mathbf{D}^{k-1},$$

où

$$\mathbf{D} = \begin{bmatrix} 1 & & & \\ & \omega & & \\ & & \ddots & \\ & & & \omega^{n-1} \end{bmatrix}.$$

Preuve:

Nous montrons tout d'abord que $\mathbf{FR} = \mathbf{DF}$.

$$\begin{aligned} \mathbf{FR}_{kj} &= \frac{1}{\sqrt{n}} (1 \omega^{k-1} \omega^{2(k-1)} \dots \omega^{(n-1)(k-1)}) \mathbf{e}_{j+1} = \frac{1}{\sqrt{n}} \omega^{j(k-1)} \\ \mathbf{DF}_{kj} &= \omega^{k-1} \mathbf{e}_k^T \frac{1}{\sqrt{n}} \begin{pmatrix} 1 \\ \omega^{j-1} \\ \omega^{2(j-1)} \\ \vdots \\ \omega^{(n-1)(j-1)} \end{pmatrix} = \frac{1}{\sqrt{n}} \omega^{k-1} \omega^{(n-1)(j-1)} = \frac{1}{\sqrt{n}} \omega^{(k-1)j}. \end{aligned}$$

$$\begin{aligned} \mathbf{FCF}^{-1} &= \mathbf{FCF}^* \\ &= \mathbf{F} (\sum_{k=1}^n c_k \mathbf{R}^{k-1}) \mathbf{F}^* = \sum_{k=1}^n c_k \mathbf{FR}^{k-1} \mathbf{F}^* \\ &= \sum_{k=1}^n c_k (\mathbf{FRF}^*)^{k-1} = \sum_{k=1}^n c_k \mathbf{D}^{k-1} = \Omega. \end{aligned}$$

Sachant que $\mathbf{D}^{k-1} = \text{diag}(1, \omega^k, \omega^{2k}, \dots, \omega^{(n-1)k})$, il en résulte que Ω est une matrice diagonale. Donc \mathbf{C} est diagonalisée par la matrice unitaire \mathbf{F} . \square

Propriété 3 Les valeurs propres sont aussi données par les éléments du vecteur \mathbf{Fc} , car $\Omega = \text{diag}(\mathbf{Fc})$.

Preuve:

En effet, l'élément Ω_{kk} est donné par

$$\begin{aligned} \Omega_{kk} &= [\sum_{k=1}^n c_k \mathbf{D}^{k-1}]_{kk} \\ &= [c_1 \mathbf{I} + c_2 \mathbf{D} + \dots + c_n \mathbf{D}^{n-1}]_{kk} \\ &= c_1 + c_2 \omega^{k-1} + \dots + c_n \omega^{(k-1)(n-1)} = [\mathbf{Fc}]_k \end{aligned}$$

5.3.1 Application 1 : propriété de la matrice \mathbf{M}_t

Nous allons appliquer l'étude précédente à la matrice twist du système 5.2. Nous déterminons, en premier, sa diagonalisation. Puis, nous déterminons ses valeurs propres et nous démontrons qu'il existe une seule valeur propre nulle lorsque n , le degré du sommet, est pair, et que lorsque n est impair, toutes les valeurs propres sont non nulles.

En effet, la matrice de twist est de la forme

$$\mathbf{M}_t = \begin{bmatrix} \frac{1}{2} & & & & & & & & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & & & & & & & \\ & & \cdot & \cdot & & & & & \\ & & & \cdot & \cdot & & & & \\ & & & & \cdot & \cdot & & & \\ & & & & & \frac{1}{2} & \frac{1}{2} & & \\ & & & & & & \frac{1}{2} & \frac{1}{2} & \end{bmatrix}.$$

Elle est donc circulante avec $c_1 = \frac{1}{2}$, $c_2 = \frac{1}{2}$, $c_3 = \dots = c_n = 0$.

La diagonalisation de \mathbf{M}_t se fait par \mathbf{F} , i.e.

$$\mathbf{F}\mathbf{M}_t\mathbf{F}^* = \sum_{k=1}^n c_k D^{k-1} = \frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{D},$$

avec $\mathbf{D} = \text{diag}(1, \omega, \dots, \omega^{n-1})$. Les valeurs propres de \mathbf{M}_t sont alors données par $v_k =$

$$\frac{1}{2} + \frac{1}{2}\omega^k, \quad k = 0, \dots, n-1$$

La matrice n'est donc pas inversible si et seulement s'il existe $k \in \{0, \dots, n-1\}$ tel que

$$\frac{1}{2} + \frac{1}{2}\omega^k = 0. \text{ Mais}$$

$$\begin{aligned} \frac{1}{2} + \frac{1}{2}\omega^{k-1} = 0 &\iff \omega^k = -1 \\ &\iff \exp\left(\frac{-2i\pi k}{n}\right) = -1 \\ &\iff k = \frac{n}{2} \\ &\iff n \text{ est pair.} \end{aligned}$$

Donc lorsque n est pair, il existe une seule valeur propre de \mathbf{M}_t qui est nulle, c'est v_p pour $p = \frac{n}{2}$. Toutes les autres valeurs propres sont non nulles. Donc l'espace propre associé (i.e. le noyau) est de dimension 1 et le rang de \mathbf{M}_t est $n-1$ si n est pair. Mais si n est impair, le rang de \mathbf{M}_t est n .

5.3.2 Application 2 : propriétés de la matrice \mathbf{M}_r

Nous appliquons maintenant les propriétés des matrices circulantes à la matrice des vecteurs tangents du système 5.1. Nous déterminons sa forme diagonalisée, et après, nous déterminons ses valeurs propres et leur multiplicité. Le lemme, ci-dessous, donne les vecteurs propres associés aux valeurs propres de \mathbf{M}_r . Nous enchaînons avec une justification du choix de Loop du vecteur $\bar{\mathbf{r}}^1$ dans le noyau de \mathbf{M}_r .

En effet, la matrice \mathbf{M}_r est de la forme

$$\mathbf{M}_r = \begin{bmatrix} \phi^0 & -\frac{1}{2} & & & & -\frac{1}{2} \\ -\frac{1}{2} & \phi^0 & -\frac{1}{2} & & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot \\ & & & & -\frac{1}{2} & \phi^0 & -\frac{1}{2} \\ -\frac{1}{2} & & & & & -\frac{1}{2} & \phi^0 \end{bmatrix}$$

Elle est donc une matrice circulante avec $c_1 = \Phi^0$, $c_2 = -\frac{1}{2}$, $c_3 = \dots = c_{n-1} = 0$, $c_n = -\frac{1}{2}$. La diagonalisation de \mathbf{M}_r donne

$$\Omega = \sum_{k=1}^n c_k D^{k-1} = \Phi^0 \mathbf{I} - \frac{1}{2} \mathbf{D} - \frac{1}{2} \mathbf{D}^{n-1},$$

d'où les valeurs propres de \mathbf{M}_r

$$\begin{aligned} v_k &= \Phi^0 - \frac{1}{2} \omega^k - \frac{1}{2} \omega^{k(n-1)}, \quad k = 0, \dots, n-1 \\ &= \Phi^0 - \frac{1}{2} \exp\left(\frac{-2i\pi k}{n}\right) - \frac{1}{2} \exp\left(\frac{-2i\pi k(n-1)}{n}\right) \\ &= \Phi^0 - \cos\left(\frac{-2\pi k}{n}\right) \end{aligned}$$

Multiplicité des valeurs propres de \mathbf{M}_r

Soient $k, l \in \{0, 1, \dots, n-1\}$, tels que $k \neq l$.

$$v_k = v_l \iff \cos\left(\frac{2\pi k}{n}\right) = \cos\left(\frac{2\pi l}{n}\right).$$

Lemme 6 v_k est de multiplicité 1 pour $k = 0$ et pour $k = \frac{n}{2}$ (si n est pair). v_k est de multiplicité 2 pour $k \neq 0$ et pour $k \neq \frac{n}{2}$. Les vecteurs propres associés sont respectivement $\mathbf{V}_0 = [1 \ 1 \ \dots \ 1]^T$, associé à v_0 ,

$\mathbf{V}_k = [1 \quad \exp(\frac{-2i\pi k}{n})^1 \quad \dots \quad \exp(\frac{-2i\pi k}{n})^{n-1}]^T$, associé à v_k pour $k \neq 0, \frac{n}{2}$,

$\mathbf{V}_k = [1 \quad (-1)^1 \quad \dots \quad (-1)^{n-1}]^T$, associé à v_k pour $k = \frac{n}{2}$ et n pair.

Notons que pour le cas où k est différent de 0 et de $\frac{n}{2}$, il y a deux vecteurs propres réels qui sont associés à v_k :

$$\mathbf{V}_k^1 = \begin{pmatrix} 1 \\ \vdots \\ \cos \frac{2\pi k(l-1)}{n} \\ \vdots \\ \cos \frac{2\pi k(n-1)}{n} \end{pmatrix} \quad \text{et} \quad \mathbf{V}_k^2 = \begin{pmatrix} 0 \\ \vdots \\ \sin \frac{2\pi k(l-1)}{n} \\ \vdots \\ \sin \frac{2\pi k(n-1)}{n} \end{pmatrix}.$$

Justification du choix de Loop

Loop choisit à une constante près $\bar{\mathbf{r}}^1 = \mathbf{U}^0 \bar{\mathbf{p}}$, avec $\mathbf{U}^0 = \left(\cos(\frac{2\pi(j-i)}{n}) \right)_{i,j=1,\dots,n}$. Ensuite, pour vérifier que ce choix appartient au noyau de \mathbf{M}_r , il montre que $\mathbf{M}_r \mathbf{U}^0 = \mathbf{0}$.

Dans le cadre général de notre étude, le noyau de \mathbf{M}_r est engendré par les vecteurs propres associés aux valeurs propres nulles de \mathbf{M}_r . En effet, Loop a choisi $\Phi^0 = \cos \frac{2\pi}{n}$, donc la seule valeur propre nulle est v_1 . Elle correspond aux deux vecteurs propres obtenus pour $k = 1$, c'est à dire les deux vecteurs \mathbf{V}_1^1 et \mathbf{V}_1^2 ci-dessus. Il suffit donc de démontrer que les vecteurs colonnes de \mathbf{U}^0 s'écrivent comme combinaisons linéaires de \mathbf{V}_1^1 et \mathbf{V}_1^2 . En effet, on a

$$\cos\left(\frac{2\pi(j-i)}{n}\right) = \cos\left(2\frac{\pi j}{n}\right) \cos\left(2\frac{\pi i}{n}\right) + \sin\left(2\frac{\pi j}{n}\right) \sin\left(2\frac{\pi i}{n}\right),$$

d'où

$$\begin{aligned} \begin{pmatrix} \cos \frac{2\pi(j-0)}{n} \\ \vdots \\ \cos \frac{2\pi(j-i)}{n} \\ \vdots \\ \cos \frac{2\pi(j-n+1)}{n} \end{pmatrix} &= \cos\left(2\frac{\pi(j-1)}{n}\right) \begin{pmatrix} 1 \\ \vdots \\ \cos \frac{2\pi(l-1)}{n} \\ \vdots \\ \cos \frac{2\pi(n-1)}{n} \end{pmatrix} + \sin\left(2\frac{\pi(j-1)}{n}\right) \begin{pmatrix} 0 \\ \vdots \\ \sin \frac{2\pi k(l-1)}{n} \\ \vdots \\ \sin \frac{2\pi k(n-1)}{n} \end{pmatrix} \\ &= \cos\left(2\frac{\pi(j-1)}{n}\right) \mathbf{V}_k^1 + \sin\left(2\frac{\pi(j-1)}{n}\right) \mathbf{V}_k^2. \end{aligned}$$

5.3.3 Caractérisation de l'espace $Im(\mathbf{M}_t)$

Les valeurs propres de \mathbf{M}_t sont $v_k = \frac{1}{2} + \frac{1}{2}\omega^{k-1}$, avec $k = 1, \dots, n$ et $\omega = \exp(\frac{-2\pi i}{n})$. Si n est impair, toutes ces valeurs propres sont non nulles. Si n est pair la seule valeur propre nulle est v_p , avec $p = \frac{n}{2}$. Donc la dimension de $Im(\mathbf{M}_t)$ (i.e. le rang de \mathbf{M}_t) est n lorsque n est impair, et $n - 1$ lorsque n est pair, i.e.

$$rg(\mathbf{M}_t) = \begin{cases} n, & \text{si } n \text{ impair,} \\ n - 1, & \text{si } n \text{ pair.} \end{cases}$$

Cela signifie que si on cherche $\bar{\mathbf{V}}$ tel que $\mathbf{M}_t \bar{\mathbf{V}} = \mathbf{V}$, alors nous aurons une solution unique lorsque n est impair, quelle que soit la valeur de \mathbf{V} . Mais si n est pair l'existence d'une solution $\bar{\mathbf{V}}$ dépendra de l'appartenance ou non du vecteur \mathbf{V} à $Im(\mathbf{M}_t)$. Il faut donc caractériser cette espace pour n pair. Supposons dans la suite de cette section que $n = 2p$.

Les vecteurs propres associés aux valeurs propres v_k , $k = 1, \dots, n$ sont donnés par les colonnes de la matrice

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^1 & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{bmatrix} \quad (5.3)$$

Afin d'alléger l'écriture, nous allons adopter la notation suivante des valeurs propres et des vecteurs propres associés : notons v_p par α_1 ($\alpha_1 = 0$) et le vecteur propre qui lui est associé par \mathbf{f}_1 . Notons aussi $v_1, \dots, v_{p-1}, v_{p+1}, \dots, v_n$ par $\alpha_2, \dots, \alpha_n$ et les vecteurs propres qui leurs sont associés par $\mathbf{f}_2, \mathbf{f}_3, \dots, \mathbf{f}_n$. L'espace image de \mathbf{M}_t est engendré par $\{\mathbf{f}_2, \mathbf{f}_3, \dots, \mathbf{f}_n\}$ (i.e. les colonnes de \mathbf{F} sauf le numéro p). Donc

$$Im(\mathbf{M}_t) = \left\{ \mathbf{V} \in \mathbb{R}^n / \mathbf{V} = \sum_{k=2}^n \lambda_k \mathbf{f}_k, \lambda_k \in \mathbb{R} \right\}$$

La proposition suivante permet de trouver $\bar{\mathbf{V}} \in \mathbb{R}^n$ tel que $\mathbf{M}_t \bar{\mathbf{V}} = \mathbf{V}$.

Proposition 3 Soit $\mathbf{V} = \sum_{k=2}^n \lambda_k \mathbf{f}_k$ un vecteur quelconque de $Im(\mathbf{M}_t)$, alors le vecteur $\bar{\mathbf{V}} = \sum_{k=1}^n \frac{1}{\alpha_k} \lambda_k \mathbf{f}_k$ vérifie

$$\mathbf{V} = \mathbf{M}_t \bar{\mathbf{V}}.$$

1. le coefficient λ_k peut même appartenir à \mathbb{R}^d , avec $d = 1, 2, 3$, car d'après la notation adoptée \mathbf{V} est un vecteur composé de n vecteurs de \mathbb{R}^d , considéré comme une matrice de dimension $n \times d$, dans ce cas, $\lambda_k \mathbf{f}_k$ est remplacé par le produit matricielle $\mathbf{f}_k \lambda_k$.

Preuve:

Puisque \mathbf{f}_k est le vecteur propre de \mathbf{M}_t associé à α_k , alors $\mathbf{M}_t \mathbf{f}_k = \alpha_k \mathbf{f}_k$, d'où

$$\begin{aligned} \mathbf{V} &= \lambda_2 \mathbf{f}_2 + \lambda_3 \mathbf{f}_3 + \dots + \lambda_n \mathbf{f}_n \\ \iff \mathbf{V} &= \frac{\lambda_2}{\alpha_2} (\alpha_2 \mathbf{f}_2) + \frac{\lambda_3}{\alpha_3} (\alpha_3 \mathbf{f}_3) + \dots + \frac{\lambda_n}{\alpha_n} (\alpha_n \mathbf{f}_n) \\ \iff \mathbf{V} &= \frac{\lambda_2}{\alpha_2} (\mathbf{M}_t \mathbf{f}_2) + \frac{\lambda_3}{\alpha_3} (\mathbf{M}_t \mathbf{f}_3) + \dots + \frac{\lambda_n}{\alpha_n} (\mathbf{M}_t \mathbf{f}_n) \\ \iff \mathbf{V} &= \mathbf{M}_t \left(\frac{\lambda_2}{\alpha_2} \mathbf{f}_2 + \frac{\lambda_3}{\alpha_3} \mathbf{f}_3 + \dots + \frac{\lambda_n}{\alpha_n} \mathbf{f}_n \right) \\ \iff \mathbf{V} &= \mathbf{M}_t \bar{\mathbf{V}}. \end{aligned}$$

□

Notons que l'espace image de la matrice \mathbf{M}_t possède une interprétation géométrique très simple. En effet, les n points du vecteur \mathbf{V} de $Im(\mathbf{M}_t)$ se trouvent sur les milieux des cotés successifs d'un polygone dont les sommets sont les n points de $\bar{\mathbf{V}}$, voir fig. 5.1.

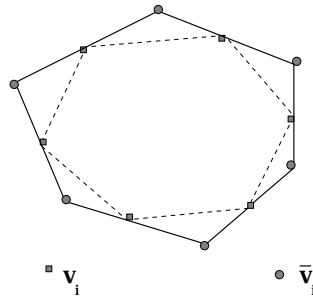


FIG. 5.1 – $\mathbf{M}_t \bar{\mathbf{V}} = \mathbf{V}$ signifie que les points \mathbf{V}_i sont les milieux des cotés successifs du polygone de sommets $\bar{\mathbf{V}}_i$.

5.3.4 Caractérisation de l'espace $\ker(\mathbf{M}_r)$

D'après le lemme 6, le noyau de \mathbf{M}_r est de dimension 2 et il est engendré par les deux vecteurs propres associés à la valeur propre nulle de \mathbf{M}_r . Ces deux vecteurs sont représentés par le vecteur d'éléments complexes

$\mathbf{V}_k = \left[1 \quad \exp\left(\frac{-2i\pi}{n}\right) \quad \dots \quad \exp\left(\frac{-2i\pi(n-1)}{n}\right) \right]^T$. On peut les écrire :

$$\mathbf{k}^a = \begin{bmatrix} 1 \\ \vdots \\ \cos \frac{2\pi(i-1)}{n} \\ \vdots \\ \cos \frac{2\pi(n-1)}{n} \end{bmatrix}, \quad \mathbf{k}^b = \begin{bmatrix} 0 \\ \vdots \\ \sin \frac{2\pi(i-1)}{n} \\ \vdots \\ \sin \frac{2\pi(n-1)}{n} \end{bmatrix}. \quad (5.4)$$

Donc tout vecteur \mathbf{V} de $\ker(\mathbf{M}_r)$ peut être exprimé dans cette base sous la forme

$$\begin{bmatrix} \mathbf{V}_k \\ \vdots \\ \mathbf{V}_k \end{bmatrix}_{n \times 3} = \begin{bmatrix} \mathbf{k}_k^a \\ \vdots \\ \mathbf{k}_k^a \end{bmatrix}_{n \times 1} (\mathbf{a})_{1 \times 3} + \begin{bmatrix} \mathbf{k}_k^b \\ \vdots \\ \mathbf{k}_k^b \end{bmatrix}_{n \times 1} (\mathbf{b})_{1 \times 3},$$

où \mathbf{a} et \mathbf{b} sont deux vecteurs quelconques de \mathbb{R}^3 . Remarquons que la multiplication matricielle avec des vecteurs de \mathbb{R}^3 est justifiée par la notation que nous avons adopté qui consiste à noter n vecteurs de \mathbb{R}^3 par une matrice de dimension $n \times 3$. Cette notation a pour but de traiter en une seule fois les trois dimensions indépendantes X , Y et Z de l'espace euclidien \mathbb{R}^3 et d'éviter d'avoir à traiter les 3 dimensions indépendamment les unes des autres.

Il est intéressant de remarquer que \mathbf{k}^a et \mathbf{k}^b appartiennent à l'espace image de \mathbf{M}_t , ce qui signifie que $\ker(\mathbf{M}_r) \subset \text{Im}(\mathbf{M}_t)$. Nous avons la proposition suivante :

Proposition 4 *Les deux vecteurs \mathbf{k}^a et \mathbf{k}^b appartiennent à l'espace $\text{Im}(\mathbf{M}_t)$, et les deux vecteurs $\bar{\mathbf{k}}^a$ et $\bar{\mathbf{k}}^b$ donnés par*

$$\bar{\mathbf{k}}^a = \begin{bmatrix} \vdots \\ \cos \frac{2\pi(j-1)}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi(j-1)}{n} \\ \vdots \end{bmatrix}, \quad \bar{\mathbf{k}}^b = \begin{bmatrix} \vdots \\ \sin \frac{2\pi(j-1)}{n} - \tan \frac{\pi}{n} \cos \frac{2\pi(j-1)}{n} \\ \vdots \end{bmatrix}$$

vérifient $\mathbf{M}_t \bar{\mathbf{k}}^a = \mathbf{k}^a$ et $\mathbf{M}_t \bar{\mathbf{k}}^b = \mathbf{k}^b$.

Preuve:

Il est clair que \mathbf{k}^a et \mathbf{k}^b sont les vecteurs propres associés à la valeur propre $\alpha_2 = \frac{1}{2} + \frac{1}{2}\omega$ de \mathbf{M}_t . Cette valeur est non nulle, donc \mathbf{k}^a et \mathbf{k}^b appartiennent à $\text{Im}(\mathbf{M}_t)$. D'après la proposition 3, le vecteur $\bar{\mathbf{f}}_2 = \frac{1}{\alpha_2} \mathbf{f}_2$, vérifie $\mathbf{M}_t \bar{\mathbf{f}}_2 = \mathbf{f}_2$, d'où

$$\bar{\mathbf{f}}_2 = \frac{2}{1 + \exp(\frac{-2\pi i}{n})} \begin{pmatrix} 1 \\ \vdots \\ \exp(\frac{-2i\pi(j-1)}{n}) \\ \vdots \\ \exp(\frac{-2i\pi(n-1)}{n}) \end{pmatrix},$$

mais

$$\begin{aligned}
\frac{2}{1+\exp(-\theta i)} \exp\left(\frac{-j\theta i}{n}\right) &= 2 \frac{\cos j\theta - i \sin j\theta}{1+\cos \theta - i \sin \theta} \\
&= 2 \frac{(\cos j\theta - i \sin j\theta)(1+\cos \theta + i \sin \theta)}{(1+\cos \theta - i \sin \theta)(1+\cos \theta + i \sin \theta)} \\
&= \frac{(1+\cos \theta) \cos j\theta + \sin \theta \sin j\theta}{1+\cos \theta} + i \frac{\sin \theta \cos j\theta - (1+\cos \theta) \sin j\theta}{1+\cos \theta} \\
&= [\cos j\theta + \tan(\theta/2) \sin j\theta] + i[\tan(\theta/2) \cos j\theta - \sin j\theta] \\
j &= 0, \dots, n-1, \quad \theta = \frac{2\pi}{n},
\end{aligned}$$

d'où

$$\bar{\mathbf{f}}_2 = \begin{pmatrix} 1 + i \tan \frac{\pi}{n} \\ \vdots \\ \cos \frac{2\pi j}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi j}{n} + i(\tan \frac{\pi}{n} \cos \frac{2\pi j}{n} - \sin \frac{2\pi j}{n}) \\ \vdots \\ \cos \frac{2\pi(n-1)}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi(n-1)}{n} + i\left(\tan \frac{\pi}{n} \cos \frac{2\pi(n-1)}{n} - \sin \frac{2\pi(n-1)}{n}\right) \end{pmatrix}$$

Or \mathbf{k}^a et \mathbf{k}^b sont respectivement la partie réelle et la partie imaginaire de \mathbf{f}_2 , donc $\bar{\mathbf{k}}^a$ et $\bar{\mathbf{k}}^b$ sont respectivement la partie réelle et la partie imaginaire de $\bar{\mathbf{f}}_2$. \square

5.4 Deuxième généralisation de la méthode

Dans cette section, nous allons utiliser les résultats de la section précédente afin de généraliser et améliorer les deux méthodes développées dans les chap. 3 et 4. L'utilisation des propriétés des deux matrices circulantes \mathbf{M}_t et \mathbf{M}_r va permettre une construction plus générale des dérivées premières et secondes des courbes frontière. L'utilisation de la caractérisation précédente des deux espaces $\text{Im}(\mathbf{M}_t)$ et $\text{ker}(\mathbf{M}_r)$ facilitera l'interprétation géométrique des paramètres de forme, ce qui aide à la modélisation de courbes frontière bien adaptées. Un réseau de courbes frontière bien adaptées permet en général de construire une surface de forme satisfaisante. Nous montrerons que cette nouvelle méthode résout le problème de twist avec une technique plus générale que celles des méthodes précédentes (les techniques du chap. 3 et de celle de Loop). Nous montrerons dans la sect. 5.4.4 que ces deux techniques sont des cas particuliers de la technique de résolution de twist avec cette méthode. La méthode utilise des facettes quintiques et suit les mêmes procédures de remplissage que celle du chap. 3. Elle laisse 6 points de contrôle

libre pour chaque macro-facette afin de pouvoir contrôler localement la forme de la surface. Plusieurs méthodes de détermination de ces points se basant sur des critères d'optimisation de forme, provenant des méthodes variationnelles de la théorie de l'élasticité seront exposées dans le chap. 6.

5.4.1 Notation

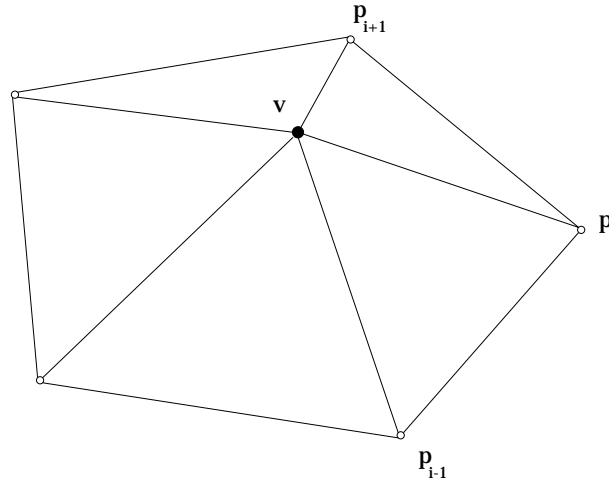


FIG. 5.2 – Notation

Soit \mathbf{v} un sommet quelconque du réseau triangulaire \mathcal{M} (en entrée) et soient \mathbf{p}_i , $i = 1, \dots, n$ les sommets voisins de \mathbf{v} , voir 5.2. Les points et les vecteurs de \mathbb{R}^3 seront notés avec des lettres en gras. Un vecteur composé de vecteurs de \mathbb{R}^3 sera noté par une lettre en gras portant un trait et désignant une matrice de dimension $n \times 3$, contenant en ces lignes des vecteurs de \mathbb{R}^3 .

Considérons la construction des n courbes frontière partant de \mathbf{v} vers \mathbf{p}_i sous les contraintes G^1 et de twist. En suivant les étapes de la méthode du chap. 4, nous nous intéressons à la construction des deux vecteurs $\bar{\mathbf{r}}^1$ et $\bar{\mathbf{r}}^2$ de dimension $n \times 3$, portant respectivement les dérivées premières et secondes en \mathbf{v} des n courbes frontière. La section 5.4.2 donne la construction de $\bar{\mathbf{r}}^1$ et la section 5.4.4 contient la construction de $\bar{\mathbf{r}}^2$

5.4.2 Nouveau calcul des dérivées premières

Les contraintes G^1 et les contraintes de twist en les sommets, voir équations 5.1 et 5.2, impliquent que le vecteur $\bar{\mathbf{r}}^1$ doit appartenir à l'intersection des deux espaces $\ker(\mathbf{M}_r)$ et $\text{Im}(\mathbf{M}_t)$ pour qu'il y ait toujours une solution. Mais la proposition 4 a montré que $\ker(\mathbf{M}_r) \subset \text{Im}(\mathbf{M}_t)$. Il suffit donc de construire $\bar{\mathbf{r}}^1$ tel qu'il appartienne à l'espace $\ker(\mathbf{M}_r)$. Dans la section 5.3.4, nous avons montré que $\ker(\mathbf{M}_r)$ est de dimension 2 et qu'il est

engendré par les deux vecteurs $\mathbf{k}^a, \mathbf{k}^b$ donnés par l'équation 5.4. Donc $\bar{\mathbf{r}}^1$ peut s'exprimer dans cette base sous la forme :

$$\begin{bmatrix} \mathbf{r}_k^1 \\ \vdots \\ \mathbf{r}_k^1 \end{bmatrix}_{n \times 3} = \begin{bmatrix} \mathbf{k}_k^a \\ \vdots \\ \mathbf{k}_k^a \end{bmatrix}_{n \times 3} (\mathbf{a})_{1 \times 3} + \begin{bmatrix} \mathbf{k}_k^b \\ \vdots \\ \mathbf{k}_k^b \end{bmatrix}_{n \times 3} (\mathbf{b})_{1 \times 3} \quad (5.5)$$

où \mathbf{a} et \mathbf{b} sont deux vecteurs de \mathbb{R}^3 . La signification géométrique de ces deux vecteurs découle facilement de la section précédente; si $\{\mathbf{a}, \mathbf{b}\}$ sont linéairement indépendants alors ils constituent une base pour le plan contenant tous les vecteurs tangents \mathbf{r}_i^1 , ce sera donc une base du plan tangent en \mathbf{v} de la surface finale, voir fig. 5.3. Les deux scalaires

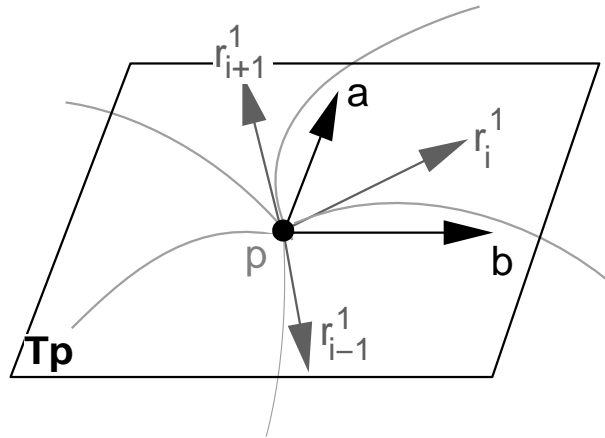


FIG. 5.3 – Interprétation géométrique des 2 vecteurs libres \mathbf{a} et \mathbf{b} dans la construction des dérivées premières.

$k_i^a = \cos \frac{2\pi(i-1)}{n}$ et $k_i^b = \sin \frac{2\pi(i-1)}{n}$ sont les coordonnées de \mathbf{r}_i^1 dans cette base.

5.4.3 Détermination pratique des vecteurs libres a et b

(1) Interpolation normale

Un vecteur normal au sommet \mathbf{v} peut être facilement interpolé, car nous pouvons choisir $\{\mathbf{a}, \mathbf{b}\}$ dans le plan passant par \mathbf{v} et perpendiculaire à la normale.

(2) Optimisation des vecteurs tangents \mathbf{r}_i^1

La configuration des vecteurs tangents des courbes frontière autour de \mathbf{v} et leurs longueurs jouent un rôle important dans la forme finale de la surface. Il faut donc choisir les meilleures valeurs possibles des vecteurs \mathbf{a} et \mathbf{b} . Pour cela, nous pouvons définir n tangentes “optimales”, notées $\bar{\mathbf{r}}_{opt}^1$, par une méthode heuristique. Ensuite on les approche

au sens des moindres carrés, i.e. les meilleures valeurs de \mathbf{a} et \mathbf{b} sont déterminées par la solution de :

$$\|\bar{\mathbf{r}}^1(\mathbf{a}, \mathbf{b}) - \bar{\mathbf{r}}_{opt}^1\|^2 \longrightarrow \min$$

La méthode heuristique que nous utilisons généralement est la méthode de Piper (voir aussi sect. 6.2.1).

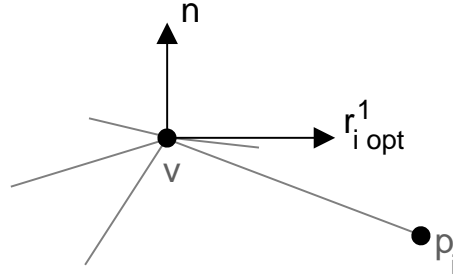


FIG. 5.4 – optimisation des dérivées premières avec une approximation aux moindres carrés de tangentes déterminées avec une méthode heuristique.

5.4.4 Nouveau calcul des dérivées secondes

La dérivée seconde $\bar{\mathbf{r}}^2$ des courbes frontière intervient dans la condition de twist, voir le système d'équations 5.2. Pour que ce système possède une solution pour les twists $\bar{\mathbf{t}}$, il faut que $\bar{\mathbf{r}}^2$ appartienne à l'espace image de la matrice de twist \mathbf{M}_t . Il doit donc exister un vecteur $\tilde{\mathbf{r}}^2 \in \mathbb{R}^{n \times 3}$ tel que $\bar{\mathbf{r}}^2 = \mathbf{M}_t \tilde{\mathbf{r}}^2$. D'après la proposition 3 si n est pair, $\bar{\mathbf{r}}^2$ peut s'exprimer dans la base des vecteurs propres de \mathbf{M}_t sous la forme suivante :

$$\bar{\mathbf{r}}^2 = \mathbf{f}_2 \lambda_2 + \dots + \mathbf{f}_n \lambda_n,$$

où $\lambda_k, k = 2, \dots, n$ sont des vecteurs de \mathbb{R}^3 et \mathbf{f}_k sont les vecteurs propres de \mathbf{M}_t donnés par les colonnes de la matrice \mathbf{F} donnée par l'équation 5.3, voir sect. 5.3.3. D'après la proposition 3 toujours, si n est pair alors le vecteur $\tilde{\mathbf{r}}^2$ est donné explicitement par

$$\tilde{\mathbf{r}}^2 = \frac{1}{\alpha_2} \mathbf{f}_2 \lambda_2 + \dots + \frac{1}{\alpha_n} \mathbf{f}_n \lambda_n,$$

où α_k sont les $n - 1$ valeurs propres (non nulles) associées respectivement aux vecteurs \mathbf{f}_k .

Donc il y a exactement $n - 1$ vecteurs libres de \mathbb{R}^3 dans le choix de $\bar{\mathbf{r}}^2$ si n est pair. Lorsque n est impair $\bar{\mathbf{r}}^2$ est complètement libre.

Ces degrés de liberté sont peu significatifs du point de vue géométrique. Nous proposons dans la suite des solutions pour la détermination pratique de ces degrés de liberté.

5.4.5 Détermination pratique des dérivées secondes

Les méthodes précédentes ([65], [84] et chap. 4) étaient totalement contraintes, on ne possédait que peu de degrés de liberté scalaires pour le choix des dérivées secondes. La nouvelle méthode que nous venons de développer dans ce chapitre permet soit un choix complètement libre des n dérivées secondes, soit un choix libre de $n - 1$ dérivées secondes en chaque sommet. Ceci est important car ainsi nous pouvons mieux contrôler la forme des courbes frontière. Nous proposons de déterminer ces vecteurs suivant une méthode heuristique combinée avec une optimisation quand le nombre de degré de liberté n'est que $n - 1$.

Nous fixons n vecteurs $\bar{\mathbf{r}}_{opt}^2$ d'après une règle heuristique que nous qualifions "d'optimales" dans le sens où nous cherchons ensuite à les approcher au mieux. Plus précisément, nous les interpolons (si n est impair) ou les approchons aux sens des moindres carrés par les $\bar{\mathbf{r}}^2(\lambda_2, \dots, \lambda_n)$ (si n est pair), i.e. avec la minimisation

$$\|\bar{\mathbf{r}}^2 - \bar{\mathbf{r}}_{opt}^2\|^2 \quad \longrightarrow \quad \min_{\lambda_k, k = 2, \dots, n.} \quad (5.6)$$

Nous avons étudié la question du choix de $\bar{\mathbf{r}}_{opt}^2$ par l'introduction de deux notions dont nous pensons qu'elles permettent de contrôler intuitivement la forme de la courbe frontière : la notion de "point cible" et de "tangente cible".

Point cible et tangente cible

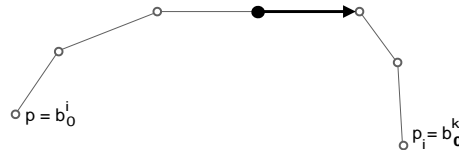


FIG. 5.5 – point cible et tangente cible.

Rappelons qu'une courbe frontière est constituée de deux morceaux cubiques se raccordant C^1 . Les deux morceaux de cubiques reliant deux sommets voisins sont construits de manière à ce que les dérivées premières et secondes en un sommet peuvent être choisies indépendamment de celles de l'autre sommet. Le schéma d'interpolation reste ainsi local. Prenons par exemple la courbe frontière allant de \mathbf{v} à \mathbf{p}_i . Appelons $\mathbf{b}_0^i, \mathbf{b}_1^i, \mathbf{b}_2^i, \mathbf{b}_3^i$ les points de Bézier du morceau construit à partir de \mathbf{v} , et appelons $\mathbf{b}_0^k, \mathbf{b}_1^k, \mathbf{b}_2^k, \mathbf{b}_3^k$ les points de Bézier du morceau construit à partir du sommet voisin \mathbf{p}_i . Le point de raccordement $\mathbf{b}_3^i = \mathbf{b}_0^k$ entre les deux morceaux cubiques se trouve obligatoirement au milieu du segment $[\mathbf{b}_2^i \mathbf{b}_2^k]$, i.e. $\mathbf{b}_3^i = \frac{1}{2}(\mathbf{b}_2^i + \mathbf{b}_2^k)$. Ce point correspond à la valeur $\frac{1}{2}$ du paramètre et appartient à la courbe. La détermination de ce point d'un vecteur tangent en ce point, permettrait

de contrôler la forme de la courbe. A cette fin, on choisit un point cible \mathbf{b}_3^{*i} et un vecteur de dérivée cible \mathbf{D}^* en ce point, voir fig. 5.5. Pour faire ce choix, nous utilisons l'une des méthodes heuristiques expliquées en sect. 6. Une fois ce choix déterminé, le point de contrôle \mathbf{b}_2^i est idéalement placé en un point

$$\mathbf{b}_2^{*i} = \mathbf{b}_3^{*i} + \frac{1}{6}\mathbf{D}^*,$$

d'où les valeurs suivantes des dérivées secondes optimales :

$$\mathbf{r}_{opt}^{2i} = 24(\mathbf{v} - 2\mathbf{b}_1^i + \mathbf{b}_2^{*i})$$

Cette technique préserve l'aspect local de la méthode tout en permettant de contrôler globalement la courbe. Ces valeurs optimales peuvent être approchées par la minimisation 5.6 ou par la minimisation de l'énergie élastique de tension de la courbe $\{\mathbf{b}_0^i, \mathbf{b}_1^i, \mathbf{b}_2^i, \mathbf{b}_3^i\}$, voir sect. 6.2.5. Notons que si les degrés de deux sommets voisins sont impair, alors le point cible et la dérivée cible seront interpolés par la minimisation 5.6, sinon ils sont approchés aux moindres carrés.

5.4.6 Vecteurs twists compatibles

Une fois les dérivées premières et secondes fixées, la deuxième partie du système twist 5.2 est dans l'espace image de \mathbf{M}_t et nous pouvons écrire

$$\Phi^1 \bar{\mathbf{r}}^1 + \Phi^0 \bar{\mathbf{r}}^2 = \mathbf{M}_t [\Phi^1 (\bar{\mathbf{k}}^a \mathbf{a} + \bar{\mathbf{k}}^b \mathbf{b}) + \Phi^0 \bar{\mathbf{r}}^2],$$

d'où la valeur explicite des vecteurs twists, solution de ce système :

$$\bar{\mathbf{t}} = \Phi^1 (\bar{\mathbf{k}}^a \mathbf{a} + \bar{\mathbf{k}}^b \mathbf{b}) + \Phi^0 \bar{\mathbf{r}}^2. \quad (5.7)$$

Cette valeur est utilisée dans l'équation 3.27 afin de trouver les points de contrôle des rubans de tangence trans-frontière le long des courbes frontière. Une valeur explicite du twist est nécessaire pour pouvoir calculer explicitement les points de contrôle de la surface.

5.4.7 Calcul des dérivées trans-frontière

Après avoir construit les courbes frontière et calculé les vecteurs twist, la méthode calcule un ruban de tangence le long de chaque courbe frontière. Le long de la courbe $M^{i-1}(0, u_i) = M^i(u_i, 0)$, on construit les deux rubans $[D_{\mathbf{u}_{i-1}} M^{i-1}] (0, u_i)$ et $[D_{\mathbf{u}_{i+1}} M^i] (u_i, 0)$. Ces dérivées assurent l'existence d'un plan commun continu le long de la courbe commune des deux facettes M^{i-1} et M^i . Elles sont construites sous les contraintes 5.1 et 5.2 de la continuité G^1 en \mathbf{v} . Elles doivent aussi interpoler les valeurs de twist aux points extrémité

de la courbe et être compatibles avec les courbes frontière déjà construites car leurs valeurs aux sommets sont imposées par les courbes frontière, voir sect. 3.9.

Une autre manière de satisfaire les conditions G^1 , différente de celle du chapitre 3, est de définir les rubans de tangence de la manière suivante :

$$\begin{cases} [D_{\mathbf{u}_{i+1}} M^i](u_i, 0) &= \Phi_i(u_i)[D_{\mathbf{u}_i} M^i](u_i, 0) + \mathbf{W}_i(u_i), \\ [D_{\mathbf{u}_{i-1}} M^{i-1}](0, u_i) &= \Phi_i(u_i)[D_{\mathbf{u}_i} M^i](u_i, 0) - \mathbf{W}_i(u_i), \end{cases} \quad (5.8)$$

où \mathbf{W}_i est une fonction vectorielle continue sur $[0, 1]$. Il est à noter que les conditions 5.8 sont équivalentes aux conditions G^1 de la section 3.2. Les conditions sur \mathbf{W}_i sont

- (1) Les valeurs $\mathbf{W}_i(0)$, $\mathbf{W}_i(1)$, et $\frac{d\mathbf{W}_i}{du_i}(0)$, $\frac{d\mathbf{W}_i}{du_i}(1)$ sont imposées par les courbes frontière et par les valeurs de twist.
- (2) \mathbf{W}_i doit être "au moins" C^0 en $u_i = \frac{1}{2}$.
- (3) \mathbf{W}_i doit être de degré ≤ 4 , pour que le degré des facettes ne dépasse pas 5.

On peut satisfaire ces trois conditions en construisant \mathbf{W}_i sous forme quadratique par morceaux interpolant aux deux extrémités les valeurs de $\mathbf{W}_i(0)$, $\mathbf{W}_i(1)$, $\frac{d\mathbf{W}_i}{du_i}(0)$ et $\frac{d\mathbf{W}_i}{du_i}(1)$. Exprimons les d'abord en fonction des courbes frontière et des twists. En effet, l'évaluation de 5.8 en $u_i = 0$ donne :

$$\mathbf{r}_{i+1}^1 = [D_{\mathbf{u}_{i+1}} M^i](0, 0) = \Phi_i(0)[D_{\mathbf{u}_i} M^i](0, 0) + \mathbf{W}_i(0) = \Phi^0 \mathbf{r}_i^1 + \mathbf{W}_i(0),$$

d'où

$$\mathbf{W}_i(0) = \mathbf{r}_{i+1}^1 - \Phi^0 \mathbf{r}_i^1. \quad (5.9)$$

Pour trouver $\frac{d\mathbf{W}_i}{du_i}(0)$, on dérive 5.8 dans la direction de l'axe des u_i et puis on évalue le résultat en $u_i = 0$, on obtient : $\mathbf{t}_i = \Phi_i'(0)\mathbf{r}_i^1 + \Phi_i(0)\mathbf{r}_i^2 + \mathbf{W}_i'(0)$,

d'où

$$\mathbf{W}_i'(0) = \mathbf{t}_i - \Phi^1 \mathbf{r}_i^1 - \Phi^0 \mathbf{r}_i^2. \quad (5.10)$$

Les valeurs de $\mathbf{W}_i(1)$ et $\mathbf{W}_i'(1)$ sont obtenues de la même manière à partir du sommet opposé \mathbf{p}_i . \mathbf{W}_i est construite sous forme quadratique par morceaux, i.e.

$$\mathbf{W}_i(u_i) = \begin{cases} \sum_{i=0}^2 \mathbf{w}_i^L B_i^2(2u), & u \in [0, \frac{1}{2}] \\ \sum_{i=0}^2 \mathbf{w}_i^R B_i^2(2u - 1), & u \in [\frac{1}{2}, 1] \end{cases} \quad (5.11)$$

où \mathbf{w}_i^L et \mathbf{w}_i^R sont respectivement les points de contrôle des deux morceaux quadratiques de \mathbf{W}_i , avec

$$\begin{aligned} \mathbf{w}_0^L &= \mathbf{r}_{i+1}^1 - \Phi^0 \mathbf{r}_i^1, \\ \mathbf{w}_1^L &= \mathbf{w}_0^L + \frac{1}{4} \mathbf{W}_i'(0) \\ &= \mathbf{r}_{i+1}^1 + \left(\frac{\Phi^1}{4} - \Phi^0\right) \mathbf{r}_i^1 + \frac{1}{4} (\mathbf{t}_i - \Phi^0 \mathbf{r}_i^2), \end{aligned}$$

et $\mathbf{w}_2^L = \mathbf{w}_2^R$ est libre, mais on choisit de prendre $\mathbf{w}_2^L = \frac{\mathbf{w}_1^L + \mathbf{w}_1^R}{2}$ afin que \mathbf{W}_i soit dans la classe C^1 sur $[0, 1]$.

La dérivée trans-frontière $[D_{\mathbf{u}_{i+1}} M^i](u_i, 0)$ est enfin obtenue en remplaçant les valeurs de $\Phi_i(u_i)$, $[D_{\mathbf{u}} M^i](u_i, 0)$ et $\mathbf{W}_i(u_i)$ dans 5.8 et en utilisant la règle de multiplication des polynômes de Bernstein :

$$B_i^p(u) B_j^q(u) = \frac{C_i^p C_j^q}{C_{p+q}^{i+j}} B_{i+j}^{p+q}(u)$$

pour calculer le produit $\Phi_i(u_i) [D_{\mathbf{u}} M^i](u_i, 0)$.

5.4.8 Remplissage des macro-facettes

Pour le remplissage des facettes quintiques, nous suivrons les mêmes procédures que celles de la section 3.10.1 et de l'annexe A.

5.4.9 Cas particuliers

Le choix particulier de Loop pour $\bar{\mathbf{r}}^1$ et $\bar{\mathbf{r}}^2$

Loop choisit à un facteur constant près $\bar{\mathbf{r}}^1 = \mathbf{U}^0 \bar{\mathbf{p}}$, avec $\mathbf{U}_{ij}^0 = \left(\cos\left(\frac{2\pi(j-i)}{n}\right) \right)$, $i, j = 1, \dots, n$. Puis il montre que $\mathbf{M}_r \bar{\mathbf{r}}^1 = \mathbf{0}$ pour vérifier que $\bar{\mathbf{r}}^1$ appartient à $\ker(\mathbf{M}_r)$.

Nous avons démontré dans le corollaire 2 de manière plus générale que $\bar{\mathbf{r}}^1$ peut s'écrire comme $\bar{\mathbf{r}}^1 = \mathbf{k}^a \mathbf{a} + \mathbf{k}^b \mathbf{b}$. $\mathbf{k}^a, \mathbf{k}^b$ désignent les deux vecteurs de $\mathbb{R}^{n \times 3}$ donnés par 5.4, constituant une base de $\ker(\mathbf{M}_r)$. \mathbf{a} et \mathbf{b} sont deux vecteurs de \mathbb{R}^3 libres. Le choix particulier de

$$\mathbf{a} = \sum_{j=1}^n \cos \frac{2\pi(j-1)}{n} \mathbf{p}_j, \quad \mathbf{b} = \sum_{j=1}^n \sin \frac{2\pi(j-1)}{n} \mathbf{p}_j. \quad (5.12)$$

à partir du voisinage du sommet correspond au choix de Loop pour $\bar{\mathbf{r}}^1$ (voir ci-dessus). Pour une explication détaillée et une interprétation géométrique de ce choix voir la section 4.3.2.

Pour le vecteur $\bar{\mathbf{r}}^2$ des dérivées secondes $\mathbf{r}_i^2 = 12(\mathbf{b}_2^i - 2\mathbf{b}_1^i + \mathbf{b}_0^i)^2$, Loop constate que $\bar{\mathbf{b}}_0$ et $\bar{\mathbf{b}}_1$ appartiennent à $\text{Im}(\mathbf{M}_t)$. Il fait donc le choix particulier de $\bar{\mathbf{b}}_2 = \frac{1}{3} \bar{\mathbf{v}} + \mathbf{U}^1 \bar{\mathbf{p}}$, avec

$$\mathbf{U}_{ij}^1 = \begin{cases} 1/6 & \text{si } j = i-1, i+1 \\ 1/3 & \text{si } j = i \\ 0 & \text{sinon.} \end{cases}$$

Ensuite, pour démontrer que $\bar{\mathbf{b}}_2$ et ainsi $\bar{\mathbf{r}}^2$ appartiennent à l'espace $\text{Im}(\mathbf{M}_t)$, il trouve le vecteur $\tilde{\mathbf{b}}_2 = \left[\frac{1}{3} \bar{\mathbf{v}} + \frac{1}{3} \mathbf{p}_i + \frac{1}{3} \mathbf{p}_{i+1} \right]_{(i=1, \dots, n)}^T$ vérifiant $\mathbf{M}_t \tilde{\mathbf{b}}_2 = \bar{\mathbf{b}}_2$. Ce choix particulier

2. le coefficient 12 est parce que Loop construit des courbes frontière quartiques.

de $\bar{\mathbf{r}}^2$ est contraint par le degré 4 des courbes frontière. En effet, le choix d'une courbe frontière en un seul morceau de degré 4 impose que le point de contrôle \mathbf{b}_2^i soit construit d'une manière symétrique des deux extrémités de la courbe frontière. Malgré le fait que ce choix vérifie les conditions de la continuité G^1 et de la compatibilité de twist, nous avons vu (sect. 2.9.1) que ce schéma appliqué à l'interpolation résulte en des ondulations indésirables des courbes frontière dues à ce choix de $\bar{\mathbf{b}}_2$.

Le cas particulier de la méthode du 4-split

La méthode du 4-split (chap. 3) résout le problème de twist par une technique similaire à celle de Loop. Le choix de $\bar{\mathbf{r}}^1$ est fait exactement de la même manière. Mais le fait que cette méthode utilise des courbes frontière cubiques par morceaux implique que le vecteur $\bar{\mathbf{b}}_2$ peut être construit d'une manière plus souple et avec suffisamment de liberté pour pouvoir éviter d'éventuelles oscillations du schéma d'interpolation. Elle fait donc le choix particulier de $\bar{\mathbf{b}}_2 = [(\gamma_0 + \gamma_1)\alpha + \frac{\gamma_2}{3}]\bar{\mathbf{v}} + \mathbf{B}^2\bar{\mathbf{p}}$ où γ_0 , γ_1 et γ_2 sont des paramètres de forme tel que $\gamma_0 + \gamma_1 + \gamma_2 = 1$. \mathbf{B}^2 est la matrice $n \times n$ donnée par la formule 3.20.

Le théorème 1 du chapitre 3 a démontré l'appartenance de $\bar{\mathbf{b}}_2$ et ainsi de $\bar{\mathbf{r}}^2$ à l'espace $Im(\mathbf{M}_t)$.

Le cas particulier de la première généralisation du 4-split

La méthode du chap. 4 est une première généralisation de la méthode du 4-split mais elle aussi est un cas particulier de la présente méthode. Les vecteurs $\bar{\mathbf{r}}^1$ et $\bar{\mathbf{r}}^2$ sont construits avec une manière similaire à celle de la méthode du 4-split, en remplaçant le voisinage réel $\bar{\mathbf{p}}$ par le voisinage virtuel $\bar{\mathbf{p}}^*$.

5.4.10 Conclusion

En s'appuyant sur les propriétés des matrices circulantes, nous avons donné dans ce chapitre une deuxième généralisation de la méthode du 4-split. Cette nouvelle méthode est strictement locale et riche de plusieurs paramètres de forme ayant des interprétations géométriques claires et permettant de contrôler et modeler localement la forme de la surface. Nous avons aussi démontré que la solution de Loop du problème de twist, présentant des défauts pour l'interpolation, peut être considérée comme un cas particulier de cette méthode. De même, la méthode de base du 4-split est aussi un cas particulier de cette méthode. La première généralisation du 4-split développée dans le chapitre précédent, a permis d'exploiter une partie seulement des caractéristiques du 4-split. Nous pensons que la dernière généralisation exploite pleinement toute la structure de la méthode du 4-split et vérifie tous les critères de performances d'un schéma d'interpolation cités dans l'introduction de cette thèse.

Dans le chapitre suivant, nous allons voir plusieurs techniques dédiées à la détermination pratique des paramètres de forme scalaires et des points de contrôle libres de cette méthode et de la méthode du chap. 4. Plusieurs méthodes heuristiques ainsi que d'autres basées sur des critères d'optimisation de forme provenant de la théorie de l'élasticité seront exposées.

Chapitre 6

Paramètres de forme et leur choix

Dans ce chapitre, nous discutons de ce qui a été le point de départ pour le développement des deux généralisations : le choix des paramètres de forme. Bien que ce choix ne soit présenté qu'en dernier chapitre, le travail de la thèse a réellement débuté avec le développement de méthodes de choix optimal des paramètres de forme pour la méthode de base du 4-split. En effet, malgré le relativement grand nombre de paramètres pour la surface issue du 4-split, il s'est avéré que des restrictions trop sévères ont été imposées aux courbes de bord. Cela a finalement initié le travail qui a été présenté dans les chapitres précédents.

Dans la suite de ce chapitre nous exposons le choix des paramètres de formes uniquement pour les deux généralisations du 4-split bien qu'une partie soit aussi applicable à la méthode de base.

Nous avons vu que les deux méthodes des deux chapitres précédents offrent un nombre de paramètres importants. D'une part, la méthode du chapitre 4 offre quatre paramètres scalaires $\alpha, \beta, \gamma_1, \gamma_2$, issus de la méthode du chapitre 3, qui permettent de contrôler localement le réseau des courbes frontière. De plus, elle introduit la notion du voisinage virtuel, qui permet à la fois de rendre la méthode strictement locale et offre, pour chaque sommet, un nombre de points libres dans le calcul des courbes frontière et des tangentes trans-frontière. D'autre part, la méthode du chapitre 5 offre encore plus de degrés de liberté. Pour chaque sommet de degré n , deux vecteurs libres permettent de choisir le plan tangent de la surface en ce sommet. De plus, n (resp. $n-1$) vecteurs libres sont disponibles pour contrôler le choix des n vecteurs des dérivées secondes des courbes frontière autour d'un sommet de degré impair (resp. pair).

Il est important de savoir utiliser ces paramètres de forme pour réellement contrôler la forme de la surface. Le choix optimal dépend beaucoup des applications. Les significations géométriques simples de la plupart des paramètres permettent de proposer à l'utilisateur plusieurs méthodes automatiques de choix de ces paramètres. Par ailleurs, ces paramètres

peuvent aussi être modifiés interactivement.

Il existe donc deux types de procédures pour un meilleur choix des paramètres de forme:

- “Design” interactif,
- Procédures automatiques qui peuvent être classées en trois groupes :
 - Méthodes heuristiques : basées sur certains choix heuristiques qui tiennent compte de la géométrie sous-jacente du maillage,
 - Procédures d’optimisation de forme : basées sur des principes issus de la théorie de l’élasticité. Ces principes permettent d’obtenir des surfaces lisses et visuellement plaisantes par l’optimisation de certaines fonctionnelles d’énergie.
 - Procédures mixtes, mélangeant un choix heuristique et une optimisation de forme.

Dans le choix d’une technique de sélection des paramètres, nous devons veiller à ce qu’elle conserve la stricte localité de l’interpolant. Nous devons par exemple appliquer localement les critères de lissage, sinon la méthode perdrait sa localité. Cette contrainte s’oppose, dans un certain sens, au but d’avoir une surface globalement lisse. Néanmoins, comme nous le verrons, nous pouvons obtenir des résultats satisfaisants.

Nous adopterons les mêmes notations que dans les chapitres précédents, et nous présenterons d’abord, sect. 6.1, les procédures du design interactif, utilisant tous les paramètres libres provenant des deux généralisations des deux chapitres précédents. Dans la sect. 6.2, nous présenterons les différentes procédures automatiques de choix des paramètres de forme provenant de la méthode du chapitre 4. La section 6.3 sera consacrée au choix automatique des paramètres de la méthode du chapitre 5.

6.1 Design interactif

Les deux méthodes des chapitres précédents ont été implémentées dans un programme interactif. Le programme prend en entrée un fichier sous format NGB (NeiGhBour), contenant la topologie et la géométrie d’un réseau surfacique (pour plus de détail sur ce format, voir annexe B). Ces données sont stockées dans trois structures principales correspondant respectivement aux sommets, aux arêtes et aux faces. Chaque structure permet d’accéder aux caractéristiques géométriques et topologiques de l’objet qu’elle représente. Après lecture du fichier NGB, un tableau est créé pour chacune de ces structures. Ces tableaux ne seront pas modifiés par la suite.

Après avoir lu les données, le programme construit la surface de la manière suivante :

1. initialisation de quelques paramètres scalaires par des valeurs prises par défaut à partir d’un fichier séparé (paramètres λ de Piper par exemple, voir sect. 6.2.1),

2. Autour de chaque sommet calculer
 - 2.1 les points de contrôle \mathbf{b}_1^i ,
 - 2.2 les points de contrôle \mathbf{b}_2^i ,
 - 2.3 les vecteurs twists compatibles,
 - 2.4 $\mathbf{V}(0)$ et $\mathbf{V}'(0)$ à partir des dérivées.
3. Parcourir l'ensemble des arêtes afin de calculer
 - 3.1 les points de contrôle au milieu \mathbf{b}_3^i ,
 - 3.2 la valeur au milieu $\mathbf{V}(\frac{1}{2})$,
 - 3.3 les points de Bézier des dérivées trans-frontière,
4. Parcourir les faces afin de
 - 4.1 élever le degré des courbes frontière de 3 à 5,
 - 4.2 élever le degré des dérivées trans-frontière de 2 à 4,
 - 4.3 remplir les patches avec les points de contrôle intérieurs.

Après le passage par ces étapes, la surface construite répond à toutes les contraintes que nous avons imposées au début (interpolation, raccordement G^1 , support local, représentation polynomiale explicite, paramètres de forme disponibles, etc). Néanmoins, il est très intéressant de pouvoir modifier cette surface interactivement afin d'obtenir des formes satisfaisantes. Le programme peut afficher la surface, sélectionner des paramètres de forme et les changer suivant le désir de l'utilisateur puis recalculer la surface autour de ce changement. Ces modifications peuvent être effectuées en temps réel, grâce à la localité de l'algorithme. On peut afficher le réseau surfacique, la surface, le réseau des courbes frontière et/ou le réseau de contrôle de Bézier de la surface. On peut faire en temps réel plusieurs transformations affines de la surface, comme la rotation, la translation et l'homothétie. Mais surtout, on dispose d'outils interactifs pour la modélisation locale de la forme de la surface. Grâce à la signification géométrique simple des paramètres de forme, l'utilisateur peut, à tout moment, changer les longueurs des tangentes et les dérivées secondes des courbes frontière autour d'un sommet sélectionné, en changeant les valeurs des paramètres scalaires $\beta, \gamma_1, \gamma_2$, ou les points libres comme le voisinage virtuel, la base du plan tangent.

La figure 6.2 montre le modèle interactif et les différents outils disponibles pour éditer la surface. La fig. 6.1 donne un exemple sur la manière d'utiliser ces outils de design : La première image montre un sommet sélectionné avec les courbes frontière qui l'entourent et leurs points de contrôle affichés en plusieurs couleurs, l'image du milieu montre certains des paramètres que nous pouvons changer autour d'un sommet : le vecteur normal, le voisinage virtuel. La troisième image montre le déplacement d'un sommet.

Rappelons encore une fois que toutes ces opérations sont locales, i.e. la surface ne sera recalculée que localement. Ceci permet une modélisation en temps réel. Nous pensons que

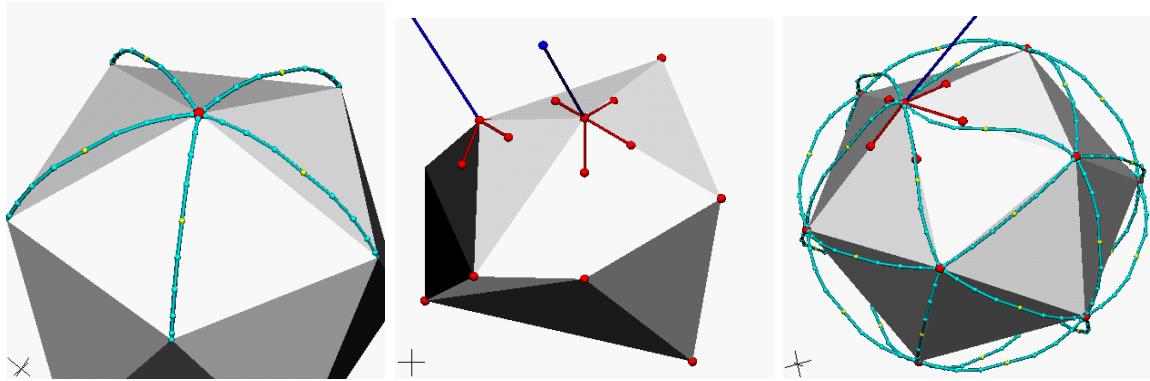
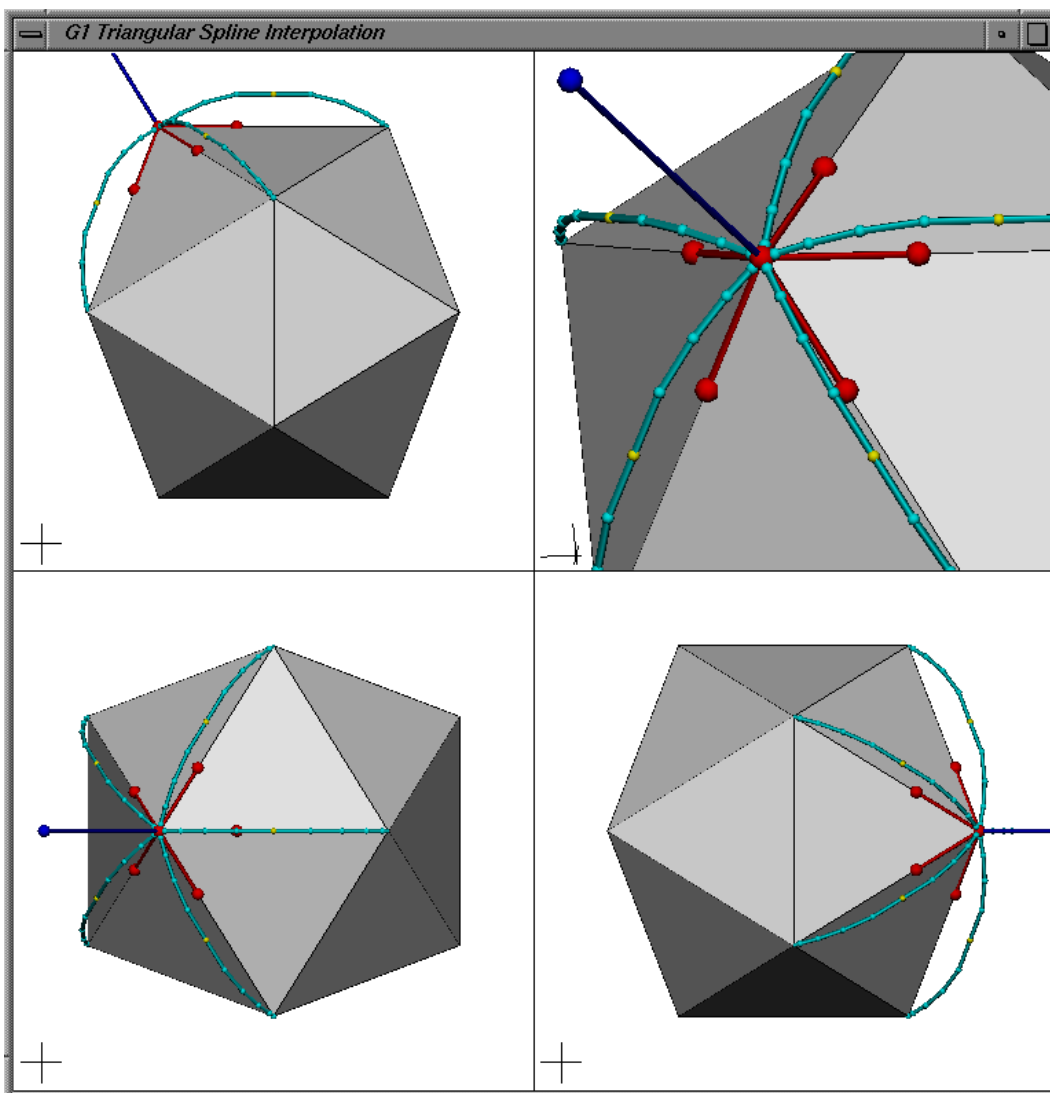


FIG. 6.1 – Outils de design interactifs.

pour des modèles de réseaux surfaciques avec un petit nombre de sommets (n'excédant pas 20 sommets), la forme désirée peut être obtenue interactivement, en jouant sur ces paramètres. Si le nombre de sommets est trop élevé, le choix interactif des paramètres libres devient trop fastidieux. Pour tout réseau surfacique il est néanmoins indispensable de disposer de procédures de choix automatique des paramètres de forme. Nous avons décidé de présenter ces procédures automatiques d'une manière cohérente avec le développement des idées de la méthode 4-split, chap. 3 et de ses deux généralisations, chap. 4 et 5.

6.2 Paramètres de forme de la première généralisation

La première généralisation de la méthode 4-split présentée dans le chap. 4 offre, pour chaque sommet \mathbf{v} de degré n , trois paramètres scalaires $\beta, \gamma_1, \gamma_2$, n points $\bar{\mathbf{p}}_i^*$ du voisinage virtuel de \mathbf{v} et six points de contrôle de la sous-facette centrale de chaque macro-facette. Nous allons présenter en 6.2.1 la sélection du paramètre β . Dans 6.2.5, la détermination de γ_1, γ_2 suivant deux approches différentes est introduite : la première utilise le concept du point et de la tangente cible, et la deuxième approche ajoute à ce concept une approche variationnelle par minimisation d'énergie. Dans 6.2.6, nous donnerons plusieurs méthodes heuristiques pour déterminer le point et la tangente cible. Nous verrons dans les sect. 6.2.3 et 6.2.4 l'utilisation du voisinage virtuel pour calculer une base optimale du plan tangent ou pour interpoler les vecteurs normaux. Enfin, les sects. 6.2.8 à 6.2.10 détailleront deux approches différentes pour la sélection des six points intérieurs de la facette centrale ; la première calcule trois points internes par une méthode heuristique basée sur une technique d'élevation de degré, puis elle calcule les trois points restant par minimisation d'énergie. La deuxième approche calcule les six points dans une seule étape de minimisation.

FIG. 6.2 – *Le modèleur interactif*

6.2.1 choix du paramètre β

Le paramètre β contrôle les longueurs des tangentes des courbes frontière qui entourent un sommet $\mathbf{v} = \mathbf{b}_0^i$, voir fin de la section 3.7.2. Des tangentes de petites longueurs produisent des facettes avec des coins pointus au voisinage de \mathbf{v} . Tandis que des tangentes trop longues peuvent causer des ondulations des facettes autour de \mathbf{v} . β est calculé suivant des considérations géométriques sur les points \mathbf{b}_1^i . En effet, une mauvaise distribution de ces points dans le plan tangent autour de \mathbf{v} peut causer des ondulations des courbes frontière. Un bon choix de ces points est celui de Piper [113]. Mais les contraintes sur les courbes frontière ne permettent pas de faire un choix *arbitraire* des tangentes. Elles

obligent les tangentes à être distribuées *régulièrement* autour d'un sommet. Cependant, nous pouvons utiliser le paramètre β pour approcher ces points au sens des moindres carrés, suivant la procédure suivante :

- projeter le point \mathbf{p}_i en un point $\tilde{\mathbf{p}}_i$ sur le plan tangent en \mathbf{v} , voir fig. 6.3,
- placer le point \mathbf{b}_1^i sur la droite définie par les deux points \mathbf{v} et $\tilde{\mathbf{p}}_i$ suivant la formule :

$$\mathbf{b}_1^{i(Piper)} = \mathbf{v} + \lambda(\tilde{\mathbf{p}}_i - \mathbf{v}).$$

Notons que Piper propose de prendre $\lambda = \frac{4}{9}\|\tilde{\mathbf{p}}_i - \mathbf{v}\|$. Nous appelons ces points **points de Piper** dans la suite.

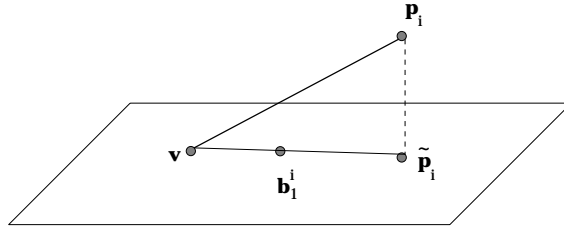


FIG. 6.3 – Construction de Piper des points \mathbf{b}_1^i .

- définir n tangentes optimales :

$$\bar{\mathbf{r}}_{opt}^1 = 6(\bar{\mathbf{b}}_1^{(Piper)} - \bar{\mathbf{b}}_0).$$

D'après la formule 4.10, les n tangentes des courbes frontière autour de \mathbf{v} sont données par

$$\mathbf{r}_i^1 = \frac{6\beta}{n} \sum_{j=1}^n \cos\left(\frac{2\pi(j-i)}{n}\right) \mathbf{p}_j$$

- déterminer β pour un λ fixé en utilisant la minimisation au sens des moindres carrés suivante :

$$\|\bar{\mathbf{r}}^1(\beta) - \bar{\mathbf{r}}_{opt}^1\|^2 = \sum_{i=1}^n \|\mathbf{r}_i^1(\beta) - \mathbf{r}_{opt_i}^1\|^2 \rightarrow \min_{\beta}.$$

Résumé: Le calcul des premières dérivées en terme des points de Bézier $\bar{\mathbf{b}}_1$ des courbes frontière a été l'objet de ce paragraphe. Pour une valeur λ fixé (Piper), le paramètre β est calculé de manière à obtenir des tangentes optimales par rapport à celle proposée par Piper.

C'est la technique qui a été développée et qui est utilisée pour la méthode de base du 4-split du chapitre 3.

6.2.2 Discussion et résultats

C'est une règle qui par défaut permet de trouver de bonnes valeurs pour le choix des tangentes. Il est évident qu'un tel choix ne permet pas de satisfaire toutes les situations. C'est ce qu'on constatera d'ailleurs aussi pour la plupart des méthodes proposées dans la suite. Illustrons l'effet de ce paramètre β sur l'exemple simple de l'icosaèdre. Fig. 6.4 montre de gauche à droite quatre surfaces interpolant l'icosaèdre pour différentes valeurs de β : 0.01, 0.12, 0.2, 0.4, les valeurs de $\lambda = 0.1$, $\gamma_0 = -3.7$, $\gamma_1 = 4.6$, $\gamma_2 = 0.1$ étant par ailleurs fixées. On peut bien voir l'effet de β sur les courbes frontières qui se tendent avec un petit β ou qui s'arrondissent pour un β optimal jusqu'à faire apparaître des ondulations indésirées des courbes frontières ce qui mène à des auto-intersections de la surface.

Cette seule étude du comportement des courbes frontière en fonction d'un scalaire β montre la nécessité d'obtenir des courbes satisfaisantes avant tout.

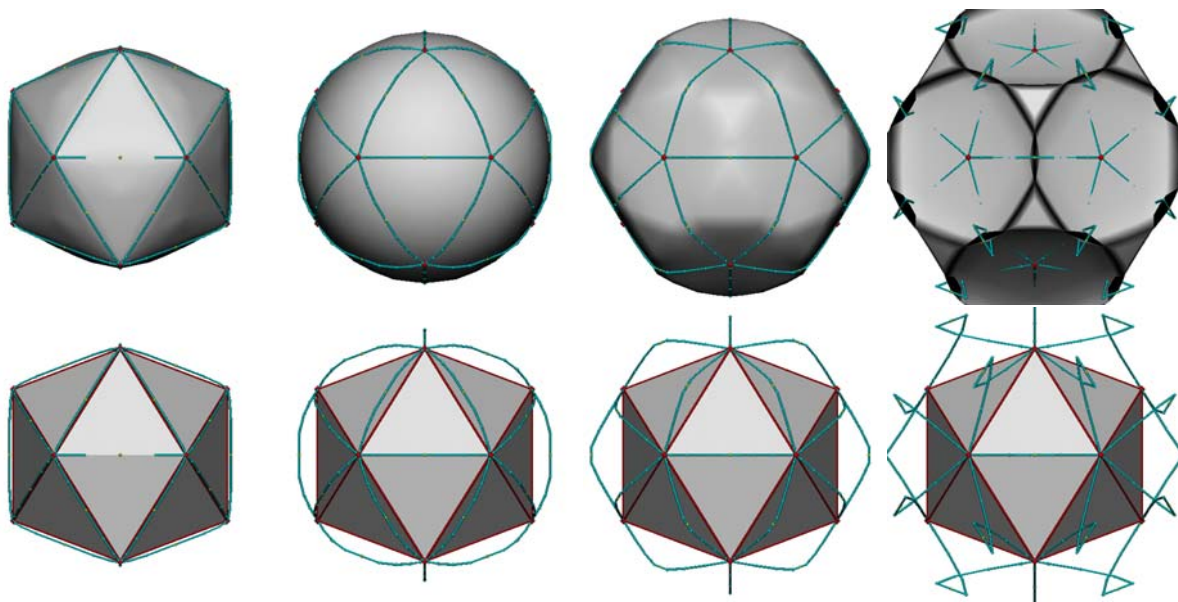


FIG. 6.4 – Interpolation de l'icosaèdre avec différentes valeurs pour le paramètre β : 0.01, 0.12, 0.2, 0.4. La ligne en haut montre la surface ensemble avec le polygone de contrôle des courbes. La ligne en bas montre les polygone de contrôle et le réseaux surfacique de l'icosaèdre à interpoler.

6.2.3 Choix du voisinage virtuel : optimisation de forme

Au lieu d'approcher les points de Piper avec les points \mathbf{b}_1^i en minimisant les distances grâce au paramètre β comme dans la section précédente, le voisinage virtuel est maintenant utilisé. Son avantage est qu'il offre plus de degrés de liberté, ainsi les points de Piper

peuvent être mieux approchés. Rappelons que la relation entre les points de contrôle \mathbf{b}_1^i et le voisinage virtuel du sommet \mathbf{v} est donnée par

$$\bar{\mathbf{b}}_1 = \alpha \bar{\mathbf{v}} + \mathbf{B}^1 \bar{\mathbf{p}}^*,$$

où $\bar{\mathbf{p}}^*$ est le voisinage virtuel de \mathbf{v} et \mathbf{B}^1 est la matrice de rang 2, donnée par 4.2.

Si l'interpolation du vecteur normal en \mathbf{v} n'est pas imposée, alors nous pouvons déterminer les points $\bar{\mathbf{p}}^*$ en résolvant le problème d'optimisation suivant :

$$\sum_{i=1}^n \|\mathbf{b}_1^i(\bar{\mathbf{p}}^*) - \mathbf{b}_1^{i(Piper)}\|^2 \rightarrow \min_{\bar{\mathbf{p}}^*}.$$

Les points \mathbf{b}_1^i ne dépendent que linéairement des points voisins virtuels \mathbf{p}_i^* . Il s'agit donc d'une minimisation au sens des moindres carrés linéaires.

6.2.4 Choix du voisinage virtuel : interpolation du vecteur normal

Nous avons vu, sect. 4.3.2, que pour interpoler le vecteur normal en \mathbf{v} , il suffit de prendre tous les points du voisinage virtuel dans le plan tangent. En pratique, nous procédons de la manière suivante : calculer d'abord les points de Piper dans le plan tangent défini par la normale donnée, suivant la méthode de la section précédente. Ensuite le voisinage virtuel est déterminé dans ce plan tangent tel que le $\bar{\mathbf{b}}_1$ approche aux moindres carrés le vecteur de $\mathbb{R}^{n \times 3}$ contenant les points de Piper.

Remarque 10 *Avec ou sans interpolation de vecteurs normaux, l'utilisation du voisinage virtuel pour approcher les points de Piper est la technique qui a été développée et qui est utilisée pour la première généralisation du 4-split présentée au chap. 4.*

6.2.5 Choix des paramètres γ_1, γ_2

Les paramètres γ_1, γ_2 sont disponibles en chaque sommet \mathbf{v} pour contrôler les dérivées secondes des courbes frontière autour de \mathbf{v} , voir sect. 3.7.3. Les points \mathbf{b}_2^i sont linéaires en γ_1, γ_2 et non pas en $\beta, \gamma_1, \gamma_2$. Pour cette raison, nous fixons la valeur de β (par la méthode du paragraphe précédent). Ainsi, les dérivées secondes en \mathbf{v} deviennent linéaires en γ_1, γ_2 et peuvent être calculées par une optimisation au sens des moindres carrés.

Notons qu'il est possible de ne pas fixer la valeur de β et d'exprimer les dérivées secondes par une fonction non linéaire de $\beta, \gamma_1, \gamma_2$ puis utiliser une procédure d'optimisation non-linéaire afin de trouver les valeurs optimales de $\beta, \gamma_1, \gamma_2$. Pour des raisons de rapidité et de convergence nous avons écarté toute méthode nécessitant des itérations non-linéaires.

Jusqu'à maintenant les valeurs de γ_1, γ_2 ont été fixées à la main. Plusieurs tests ont montré que les

$$\text{valeurs par défaut : } \quad \gamma_1 = 4.6 \quad \gamma_2 = 0.1 \quad (6.1)$$

donnent des résultats très satisfaisants quand le réseau surfacique est à peu près régulier. Pour un bon fonctionnement, un choix automatique de ces paramètres en fonction du réseau à interpoler semble indispensable. Nous allons proposer deux approches différentes.

Approche 1 : approcher point et tangente cibles

Pour déterminer γ_1, γ_2 , nous proposons d'approcher les segments des courbes cubiques, définis sur $[0, \frac{1}{2}]$, par leurs points de contrôle $\mathbf{b}_0^i, \mathbf{b}_1^i, \mathbf{b}_2^{opt_i}, \mathbf{b}_3^{*i}$, où \mathbf{b}_3^{*i} est un point cible qu'on souhaite atteindre par la courbe frontière allant de \mathbf{v} à \mathbf{p}_i , voir sect. 5.4.5 et sect. 6.2.6. Le point $\mathbf{b}_2^{opt_i}$ est défini par

$$\mathbf{b}_2^{opt_i} = \mathbf{b}_3^{*i} + \frac{1}{6}\mathbf{D}_i^*,$$

avec \mathbf{D}_i^* est une dérivée cible qu'on souhaite atteindre en \mathbf{b}_3^i par la dérivée de la courbe frontière allant de \mathbf{v} à \mathbf{p}_i . Nous pouvons résoudre aux sens des moindres carrés le problème suivant :

$$\|\bar{\mathbf{b}}_2(\gamma_1, \gamma_2) - \bar{\mathbf{b}}_2^{opt}\|^2 = \sum_{i=1}^n \|\mathbf{b}_2^i(\gamma_1, \gamma_2) - \mathbf{b}_2^{opt_i}\|^2 \rightarrow \min_{\gamma_1, \gamma_2}.$$

Le point $\bar{\mathbf{b}}_2$ en fonction de γ_1, γ_2 est donné par 4.1, sect 4.1. Notons que ce problème d'optimisation peut aussi s'écrire de manière équivalente :

$$\|\bar{\mathbf{r}}^2(\gamma_1, \gamma_2) - \bar{\mathbf{r}}_{opt}^2\|^2 \rightarrow \min_{\gamma_1, \gamma_2}, \quad (6.2)$$

avec

$$\begin{aligned} \bar{\mathbf{r}}_{opt}^2 &= 24(\mathbf{v} - 2\bar{\mathbf{b}}_1 + \bar{\mathbf{b}}_2^{opt}), \\ \bar{\mathbf{r}}^2 &= 24(\mathbf{v} - 2\bar{\mathbf{b}}_1 + \bar{\mathbf{b}}_2). \end{aligned} \quad (6.3)$$

Tout dépend du "bon" choix de ces éléments "cible" que l'on essaie d'approcher.

Approche 2 : minimisation de l'énergie de tension des courbes frontière

Pour construire des courbes frontière esthétiques et qui évitent les ondulations une approche variationnelle s'impose. Ces méthodes font l'objet du domaine intitulé "Variational Design" en CAGD. Ne citons que les travaux les plus connus [105, 64, 20, 19], etc. Nous utilisons l'intégrale d'énergie de tension généralisée qui consiste à combiner l'énergie de tension classique avec un contrôle sur la longueur d'arc ([121], [99], [112], [17]) en version linéarisée comme suivant :

$$\sigma(\beta, \gamma_1, \gamma_2) = \sum_{i=1}^n \left[\int_0^{\frac{1}{2}} \|X_i''(t)\|^2 dt + \omega^2 \int_0^{\frac{1}{2}} \|X_i'(t)\|^2 dt \right], \quad (6.4)$$

où $X_i(t)$ est une courbe cubique définie sur $[0, \frac{1}{2}]$, par les points de contrôle $\{\mathbf{v}, \mathbf{b}_1^i, \mathbf{b}_2^i(\gamma_1, \gamma_2), \mathbf{b}_3^{*i}\}$, et ω est un paramètre de lissage, permettant de courber ou de tendre les courbes, i.e. de contrôler leur longueur. Notons que nous avons remplacé \mathbf{b}_3^i par le point cible \mathbf{b}_3^{*i} afin d'avoir un critère de lissage local et ne pas perdre la localité de la méthode.

En remplaçant

$$X_i(t) = \sum_{j=0}^3 \mathbf{b}_j^i B_j^3(t)$$

dans l'équation 6.4, nous obtenons :

$$\sigma(\gamma_1, \gamma_2) = \sum_{i=0}^3 \sum_{j=0}^3 (\bar{\mathbf{b}}_i \cdot \bar{\mathbf{b}}_j) \int_0^{\frac{1}{2}} (B_i''(t)B_j''(t) + \omega^2 B_i'(t)B_j'(t)) dt, \quad (6.5)$$

puis, en remplaçant $\bar{\mathbf{b}}_0, \bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \bar{\mathbf{b}}_3$ comme définis par l'équation 4.8, nous obtenons enfin l'expression suivante pour l'énergie de tension généralisée σ qui est quadratique en γ_1, γ_2 :

$$\sigma(\gamma_1, \gamma_2) = \frac{1}{2} \mathbf{\Gamma}^T M \mathbf{\Gamma} - \mathbf{L}^T \mathbf{\Gamma} + k, \quad (6.6)$$

avec $\mathbf{\Gamma} = (\gamma_1, \gamma_2)^T$, M une matrice (2×2) symétrique, \mathbf{L} un vecteur (2×1) et k une constante. σ est donc une fonction quadratique convexe en $\mathbf{\Gamma}$, son minimum est atteint lorsque le gradient est nul, i.e. $(\gamma_1, \gamma_2)^T$ est la solution du système linéaire :

$$M \mathbf{\Gamma} = \mathbf{L}.$$

On applique la même méthode sur le deuxième segment de la courbe et ainsi nous obtenons le point \mathbf{b}_2^k , avec k l'indice de \mathbf{v} par rapport à \mathbf{p}_i . Le point milieu \mathbf{b}_3^i , effectuant le joint du segment, est enfin choisi au milieu du segment $[\mathbf{b}_2^i \mathbf{b}_2^k]$. Il approche seulement le point cible \mathbf{b}_3^{*i} .

6.2.6 Méthodes heuristiques pour la construction du point et de la tangente cible

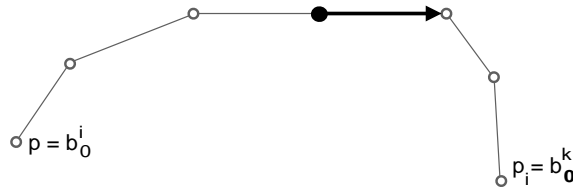


FIG. 6.5 – Construction d'un point et d'une tangente cible.

L'introduction d'un point cible et d'une tangente cible entre deux sommets adjacents du réseau surfacique est une manière intuitive de contrôler la forme de la courbe

frontière, fig 6.5. Les courbes frontière sont construites morceau par morceau de telle manière que les points \mathbf{b}_3^i et la tangente en ce point milieu approchent ou interpolent les points et tangentes cibles. Le choix de ces points et tangentes cibles est contraint par deux conditions :

- (1) lissage de la forme finale de la courbe entière,
- (2) l'interpolant doit rester locale.

La première condition est très subjective et dépend beaucoup des applications. L'utilisateur choisit le point et la tangente cible suivant une méthode heuristique qui dépend de la topologie du réseau surfacique qu'il souhaite représenter par une surface lisse. La deuxième condition insiste sur la nature locale de l'interpolant. Dans la suite, nous allons proposer quelques schémas heuristiques, basés sur la géométrie sous-jacente du réseau surfacique, pour la construction de ces cibles. Notons que ces cibles peuvent être interpolées lorsque la deuxième généralisation du 4-split est utilisée et lorsque les degrés des deux sommets extrêmes de la courbe sont impairs.

• Point cible par subdivision “4-point”

Considérons un sommet \mathbf{v} voisin à n sommets \mathbf{p}_i , $i = 1, \dots, n$. Pour la courbe frontière $X_i(t)$, allant de \mathbf{v} à \mathbf{p}_i , les deux premiers et les deux derniers points de contrôle $\mathbf{b}_0^i, \mathbf{b}_1^i, \tilde{\mathbf{b}}_0^i, \tilde{\mathbf{b}}_1^i$ de chacun de ces deux morceaux cubiques ont été fixés. Nous déterminons maintenant le point de contrôle cible en fonction de ces 4 points en appliquant la règle de subdivision du *4-point scheme* de Dyn, Levin et Gregory [34], [32] :

$$\mathbf{b}_3^{*i} = \frac{1}{16}(-\mathbf{b}_0^i + 9\mathbf{b}_1^i + 9\tilde{\mathbf{b}}_1^i - \tilde{\mathbf{b}}_0^i)$$

Cette formule consiste à interpoler les quatre points par un polynôme cubique et une paramétrisation équidistante, à évaluer cet interpolant au milieu de l'intervalle des paramètres.

• Point cible par subdivision “butterfly” simplifiée

La règle précédente ne tient que compte des informations sur l'arête sous-jacente et du plan tangent en les extrémités. Une autre approche issue des schémas de subdivision consiste à tenir compte de la taille et de la position des deux triangles voisins de l'arête qui définit la courbe. Le point cible est déterminé par une simplification du *Butterfly scheme* [32] qui n'utilise que les quatre points des deux triangles partageant l'arête \mathbf{vp}_i , fig.6.6 :

$$\mathbf{b}_3^{*i} = \frac{1}{4}(-\mathbf{v} - \mathbf{p}_i + 3\mathbf{p}_{i-1} + 3\mathbf{p}_{i+1}).$$

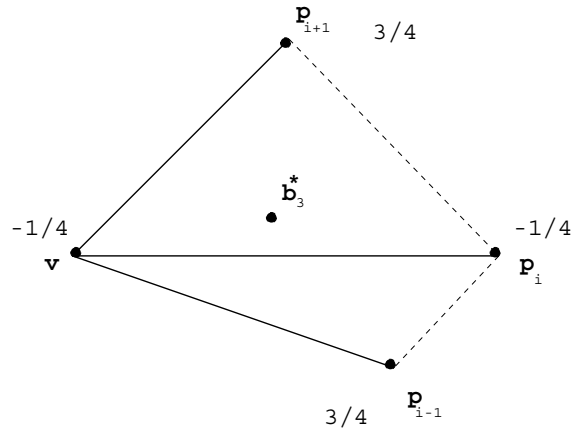


FIG. 6.6 – Point cible facial.

- **Tangente cible**

Nous proposons de prendre la tangente cible au point milieu de la courbe allant de \mathbf{v} à \mathbf{p}_i , comme la moitié du vecteur $\mathbf{p}_i\vec{\mathbf{v}}$, i.e.

$$\mathbf{D}_i^* = \frac{1}{2}(\mathbf{v} - \mathbf{p}_i),$$

car idéalement la courbe devrait suivre l'arête qui lui correspond. La longueur de cette tangente peut néanmoins rester variable.

Résumé: *Le calcul des dérivées secondes en terme des points de Bézier $\bar{\mathbf{b}}_2$ des courbes frontière a été l'objet de ce paragraphe. Les paramètres scalaires γ_1, γ_2 sont les degrés de liberté pour le calcul de ces dérivées secondes autour d'un sommet. Deux pistes ont été explorées. La première définit des points et tangentes cibles au milieu de la courbe frontière entière et essaie de les approcher au mieux. La seconde approche détermine ces dérivées secondes par minimisation de l'énergie de tension sur chaque morceau de courbe.*

6.2.7 Discussion et résultats

Toutes les méthodes qui viennent d'être présentées ont pour but de construire automatiquement des *courbes frontière lisses* en connaissant uniquement le réseau surfacique en entrée. En pratique nous utilisons l'optimisation sur le voisinage virtuel en approchant les points "idéaux" de Piper. Si la normale n'est pas donnée, alors un autre vecteur normal est estimé à partir du réseau afin de calculer les points de Piper. Il faut noter que le plan tangent de la surface n'est pas nécessairement défini par ce dernier vecteur normal, mais il est exactement donné par la formule 4.10. Les paramètres γ_1, γ_2 sont ensuite déterminés

en minimisant l'énergie de tension. Cette approche peut, en général, produire des courbes plus lisses par rapport à la première approche. L'effet de la minimisation d'énergie est illustré en fig. 6.7. A gauche, des valeurs par défaut $\gamma_1 = 4.6, \gamma_2 = 0.1$ (voir 6.1) ont été utilisées. A droite, ces paramètres ont été déterminés par minimisation d'énergie. On voit ici que les légères ondulations des deux courbes disparaissent.

Il est à noter que cette méthode ne minimise pas exactement l'énergie de la courbe. En effet, l'intégrale 6.4 n'est qu'une simplification de cette énergie. On doit intégrer par rapport à l'abscisse curviligne pour obtenir l'énergie exacte. Mais cette méthode de simplification a été adoptée par plusieurs auteurs et a montrée son efficacité (voir par exemple [21, 116]). D'autre part, les courbes de bords sont toutes soumises à de sévères contraintes concernant les tangentes. Il y a très peu de liberté sur elles. Plus particulièrement, des ondulations des courbes dues aux tangentes ne peuvent pas toujours être rattrapées par une minimisation d'énergie sur les dérivées secondes. Plus de détails à ce sujet seront donnés à propos de la deuxième généralisation du 4-split en sect. 6.3.

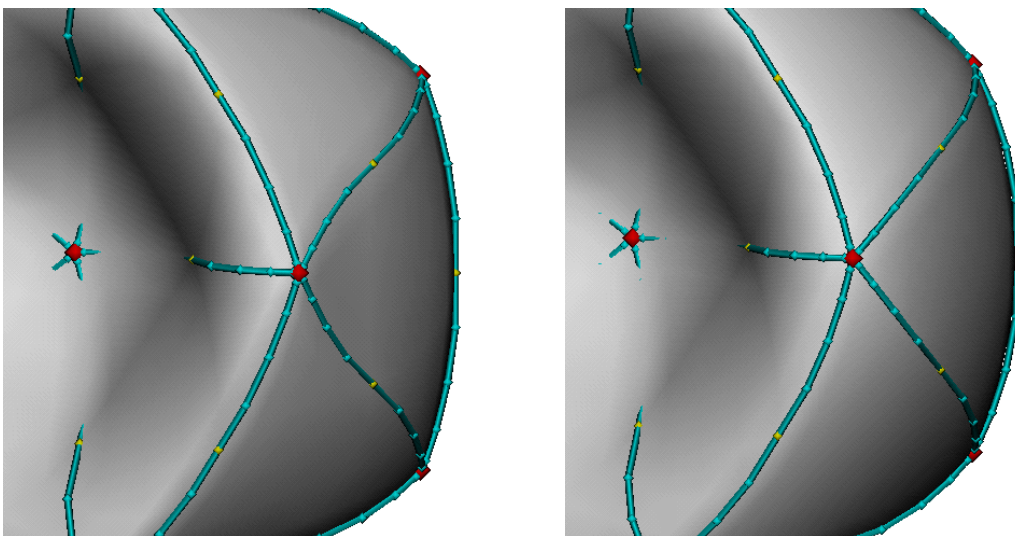


FIG. 6.7 – *Choix de γ_1, γ_2 : Interpolation d'un réseau non convexe. A gauche : valeurs par défaut : $\gamma_1 = 4.6, \gamma_2 = 0.1$. A droite : minimisation de l'énergie de tension.*

6.2.8 Choix des points de contrôle intérieurs

Après avoir fixé entièrement les courbes frontière, les deux rangées de points de contrôle des macro-facettes sont ainsi déterminés. Le raccord de continuité G^1 entre les macro-facettes est donc assuré. Il reste à remplir les macro-facettes, i.e. à déterminer les 15 points de contrôle non calculés. Nous adoptons maintenant la notation de la figure 6.8,

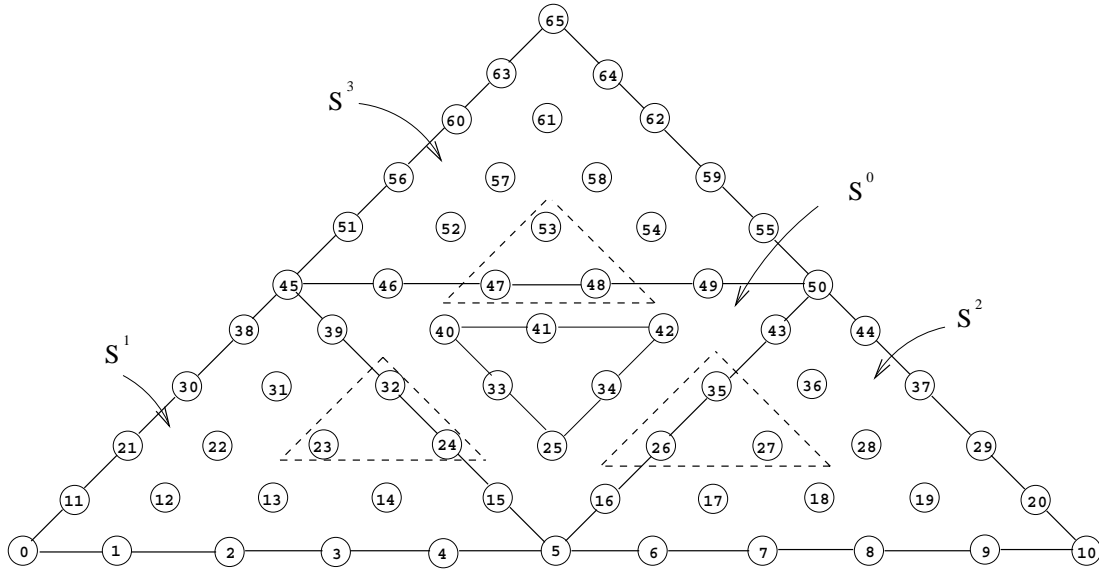


FIG. 6.8 – Notation des points de Bézier d'une macro-facette.

où les 66 points de contrôle d'une macro-facette sont désignés par un entier.

Les 15 points inconnus peuvent être partagés en deux ensembles P_1 et P_2 de la manière suivante: (voir fig. 6.8)

$$\begin{aligned} P_1 &= \{23, 32, 24, 27, 26, 35, 53, 47, 48\}, \\ P_2 &= \{25, 33, 34, 40, 41, 42\}. \end{aligned}$$

Rappelons que nous avons imposé une continuité C^1 le long des courbes internes de la macro-facette. La proposition suivante soutient ce partage des points :

Proposition 5 *Si les points de l'ensemble P_2 sont fixés alors les points de l'ensemble P_1 sont imposés par les conditions continuité C^1 le long des 3 courbes internes de la macro-facette. (Cf. sect. 1.2.5)*

Preuve:

Les conditions de continuité C^1 le long des trois courbes internes que nous avons

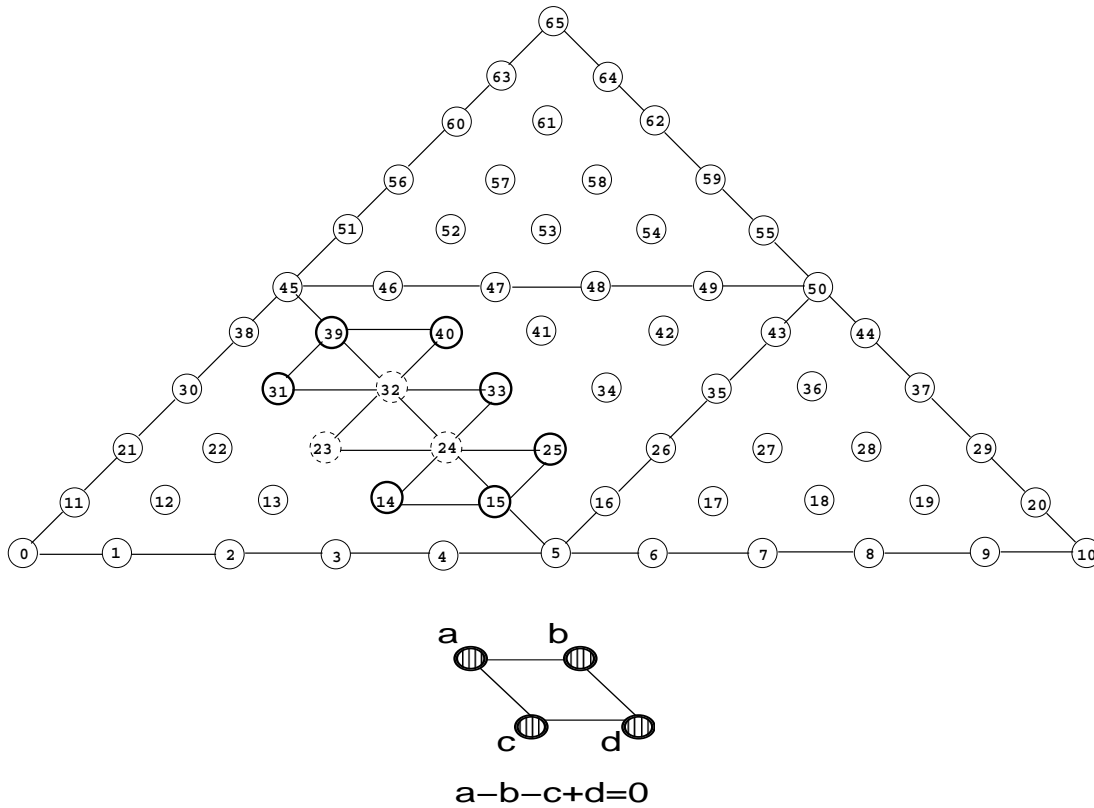


FIG. 6.9 – Conditions de parallélogrammes le long de la courbe interne allant du point 5 au point 45 : les trois points 23, 24, 32 sont obtenus des 3 parallélogrammes montrés.

appelées conditions de parallélogrammes s'écrivent (voir 6.9) :

$$\left\{ \begin{array}{l}
 14 - 24 - 15 + 25 = 0 \\
 23 - 32 - 24 + 33 = 0 \\
 31 - 39 - 32 + 40 = 0 \\
 \\
 52 - 46 - 47 + 40 = 0 \\
 53 - 47 - 48 + 41 = 0 \\
 54 - 48 - 49 + 42 = 0 \\
 \\
 36 - 43 - 35 + 42 = 0 \\
 27 - 35 - 26 + 34 = 0 \\
 17 - 26 - 16 + 25 = 0
 \end{array} \right. \quad (6.7)$$

Ce système linéaire peut s'écrire matriciellement :

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 23 \\ 24 \\ 32 \\ 53 \\ 47 \\ 48 \\ 27 \\ 35 \\ 26 \end{bmatrix} = \begin{bmatrix} 14 + 15 - 25 \\ -33 \\ 39 - 31 - 40 \\ 46 - 40 - 52 \\ -41 \\ 49 - 42 - 54 \\ 43 - 36 - 42 \\ -34 \\ 16 - 17 - 25 \end{bmatrix}.$$

Or, le déterminant de ce système linéaire est non nul, donc les points de l'ensemble P_1 sont bien fixés par ceux de l'ensemble P_2 . \square

On déduit de cette proposition qu'il y a seulement **6 degrés de liberté vectoriels**, à savoir les six points de contrôle internes de la sous-facette centrale.

En pratique, ces points sont des paramètres de forme et permettent de contrôler la forme de la macro-facette. Pour les déterminer, nous proposons deux approches différentes : la première (sect. 6.2.9) calcule d'abord les trois points internes 33, 34, 41 par une méthode heuristique basée sur une technique d'élevation de degré. Ensuite, les points twists 25, 40, 42 sont calculés par une minimisation d'énergie. La deuxième approche (sect. 6.2.10) calcule à la fois les 6 points de P_2 par une minimisation d'énergie.

6.2.9 Première approche : choisir les trois points internes de S^m par une technique d'élevation de degré

On calcule dans un premier temps les trois points internes 33, 34, 41 de la facette centrale par une méthode heuristique. On peut résumer cette approche en deux étapes principales.

Première étape

Nous exposons seulement la construction du point 33, car les deux autres points 34 et 41 se déterminent de manière symétrique.

Pour cela, considérons la courbe interne C allant du point **5** au point **45**. Notons par \mathbf{A}_i , $i = 0, 1, 2, 3, 4, 5$ les points de contrôle de C et par \mathbf{B}_i , $i = 0, 1, 2, 3, 4$ les points de contrôle se situant immédiatement à gauche de C , voir fig. 6.10. Introduisons aussi les vecteurs suivants

$$\begin{aligned} \mathbf{U}_i &= \mathbf{B}_i - \mathbf{A}_i, \\ \tilde{\mathbf{U}}_i &= \mathbf{B}_i - \mathbf{A}_{i+1}, \end{aligned} \quad i = 0, 1, 3, 4.$$

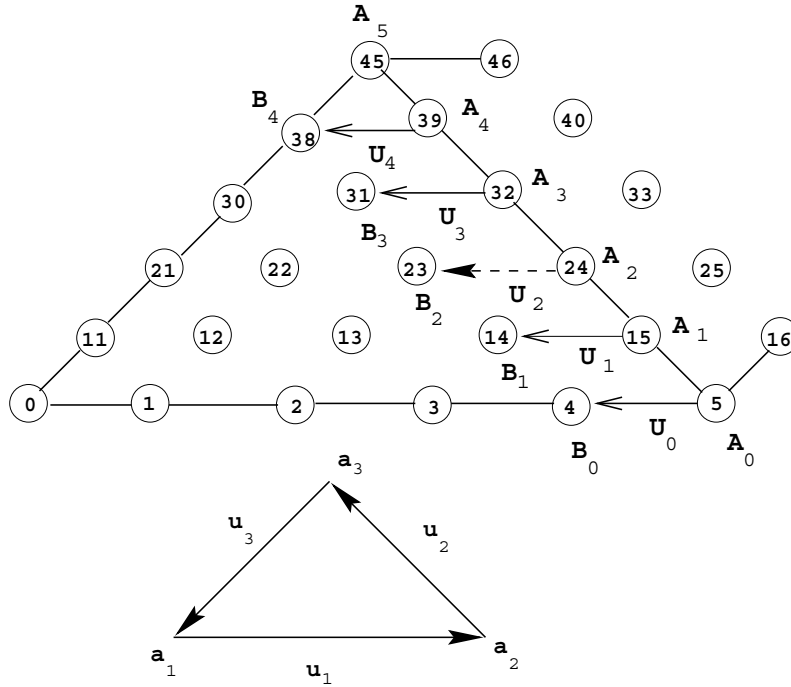


FIG. 6.10 – Méthode heuristique pour le calcul des points de contrôle interne 33, 34, 41.

Avec ces notations, les dérivées trans-frontière de la sous-facette S^1 le long de la courbe C dans les directions $-\mathbf{u}_1$ et $-\mathbf{u}_3$ sont données par : (voir fig. 6.10)

$$\begin{aligned} -S_{\mathbf{u}_1}^1(0, 1-t, t) &= 5 \sum_{i=0}^4 \mathbf{U}_i B_i^4(t), \\ -S_{\mathbf{u}_3}^1(0, 1-t, t) &= 5 \sum_{i=0}^4 \tilde{\mathbf{U}}_i B_i^4(t), \quad \forall t \in [0, 1]. \end{aligned}$$

Les deux vecteurs \mathbf{U}_2 et $\tilde{\mathbf{U}}_2$ sont inconnus. Pour les définir, nous supposons que les vecteurs $\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_3, \mathbf{U}_4$ proviennent d'une courbe de Bézier cubique élevée en degré. Le point \mathbf{U}_2 supposé appartenir à la courbe élevée en degré 4 (sect. 1.1.3) est donc donnée par la formule :

$$\mathbf{U}_2 = \frac{1}{6}(-\mathbf{U}_0 + 4\mathbf{U}_1 + 4\mathbf{U}_3 - \mathbf{U}_4).$$

On procède de manière analogue pour l'autre dérivée transfrontière de S^1 avec les vecteurs de contrôle $\tilde{\mathbf{U}}_0, \tilde{\mathbf{U}}_1, \tilde{\mathbf{U}}_3, \tilde{\mathbf{U}}_4$ pour obtenir après une élévation de degré le vecteur de contrôle :

$$\tilde{\mathbf{U}}_2 = \frac{1}{6}(-\tilde{\mathbf{U}}_0 + 4\tilde{\mathbf{U}}_1 + 4\tilde{\mathbf{U}}_3 - \tilde{\mathbf{U}}_4).$$

Nous obtenons alors deux points candidats pour le point \mathbf{B}_2 : $\mathbf{B}_2^1 = \mathbf{A}_2 + \mathbf{U}_2$ et $\mathbf{B}_2^2 = \mathbf{A}_3 + \tilde{\mathbf{U}}_2$. Nous choisissons \mathbf{B}_2 comme la moyenne des deux points, i.e.

$$\begin{aligned} \mathbf{B}_2 &= \frac{1}{2}(\mathbf{B}_2^1 + \mathbf{B}_2^2) \\ &= \frac{1}{2}(\mathbf{A}_2 + \mathbf{A}_3 + \mathbf{U}_2 + \tilde{\mathbf{U}}_2). \end{aligned} \tag{6.8}$$

Le point 33 est calculé par le parallélogramme reliant les points 23(\mathbf{B}_2), 32, 33, 24, i.e. par la formule :

$$33 = 32 + 24 - 23.$$

Les deux points 34 et 41 sont calculés par la même méthode appliquée aux deux autres courbes internes.

Remarque 11 *Il est important de remarquer, pour l'étape de minimisation suivante, qu'il existe maintenant une dépendance linéaire des trois points restant inconnus, 25, 40 et 42 en fonction de tous les autres points.*

Deuxième étape

La deuxième étape de cette approche consiste à calculer les trois points twists 25, 40 et 42 par la minimisation de l'énergie donnée par l'une des deux fonctionnelles quadratiques suivantes :

$$E_k(X, Y) = \int_{\Delta} \left(\frac{\partial^k X}{\partial u^k} \cdot \frac{\partial^k Y}{\partial u^k} + \frac{\partial^k X}{\partial v^k} \cdot \frac{\partial^k Y}{\partial v^k} + \frac{\partial^k X}{\partial w^k} \cdot \frac{\partial^k Y}{\partial w^k} \right) dudvdw, \quad (6.9)$$

avec $k = 2$ ou 3 et X, Y deux fonctions définies sur Δ .

Nous déterminons alors les points 25, 40 et 42 de manière à minimiser

$$\sigma = \sum_{j=0}^3 E_k(\mathbf{S}^j, \mathbf{S}^j) \quad (6.10)$$

où $\mathbf{S}^j = \sum_{|i|=5} \mathbf{s}_i^j B_i^5(u, v, w)$ sont les 4 sous-facettes composant une macro-facette quelconque, avec $\mathbf{S}^0 = \mathbf{S}^m$ la sous-facette centrale.

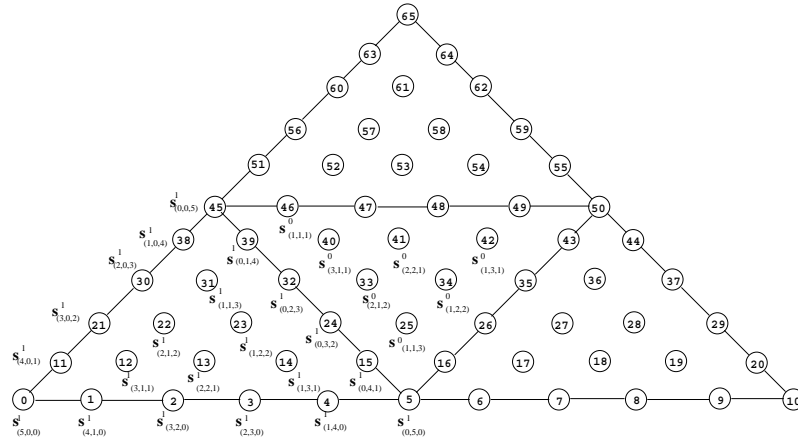
Écrivons d'abord l'équation 6.10 sous forme matricielle. Pour cela, posons les 3 points inconnus de S^m dans un vecteur (voir fig. 6.11)

$$\bar{\mathbf{s}} = [\mathbf{s}_{113}^0, \mathbf{s}_{311}^0, \mathbf{s}_{131}^0]^T = [25, 40, 42]^T.$$

Lemme 7 *L'équation 6.10 s'écrit*

$$\sigma(\bar{\mathbf{s}}) = \sum_{j=0}^3 [M^j \bar{\mathbf{s}} + \mathbf{a}^j]^T A^k [M^j \bar{\mathbf{s}} + \mathbf{a}^j],$$

où M^j , $j = 0, 1, 2, 3$ sont trois matrices 21×3 , \mathbf{a}^j sont trois vecteurs (21×3) ne dépendant pas des trois points $\mathbf{s}_{113}^0, \mathbf{s}_{311}^0, \mathbf{s}_{131}^0$ et A^k est une matrice (21×21).

FIG. 6.11 – Notation des points de contrôle entrant dans $E_k(\mathbf{S}^1, \mathbf{S}^1)$ **Preuve:**

Démontrons que $E_k(\mathbf{S}^1, \mathbf{S}^1) = (M^1 \bar{\mathbf{s}} + \mathbf{a}^1)^T A_k (M^1 \bar{\mathbf{s}} + \mathbf{a}^1)$. Commençons d'abord par organiser les points de contrôle de S^1 suivant l'ordre

$$I_5 = \{(i, j, k), \quad k = 0, \dots, 5 \quad i = 0, \dots, 5 - k, \quad j = 0, \dots, 5 - k - i, \text{ et } i + j + k = 5\}.$$

Donc les points de contrôle de S^1 peuvent s'écrire :

$$\begin{aligned} \mathbf{s}^1 &= \left[\mathbf{s}_{(i,j,k)}^1, \quad (i, j, k) \in I_5 \right]^T \\ &= \left[\mathbf{s}_{5,0,0}^1, \mathbf{s}_{4,1,0}^1, \dots, \mathbf{s}_{0,0,5}^1 \right]^T. \end{aligned}$$

En posant $A_{\mathbf{ij}}^k = E_k(B_{\mathbf{i}}^5, B_{\mathbf{j}}^5)$, avec $\mathbf{i}, \mathbf{j} \in I_5$, $E_k(\mathbf{S}^1, \mathbf{S}^1)$ s'écrit

$$\begin{aligned} E_k(\mathbf{S}^1, \mathbf{S}^1) &= \sum_{\mathbf{i} \in I_5} \sum_{\mathbf{j} \in I_5} A_{\mathbf{ij}}^k \mathbf{s}_{\mathbf{i}}^1 \cdot \mathbf{s}_{\mathbf{j}}^1, \\ &= [\mathbf{s}^1]^T A^k [\mathbf{s}^1], \end{aligned} \tag{6.11}$$

Les points de contrôle de S^1 dépendant des trois points inconnus sont au nombre de trois : il s'agit de $s_{(1,2,2)}^1, s_{(0,2,3)}^1, s_{(0,3,2)}^1$. Leurs relations avec les points twist sont données par 6.7 et 6.8, i.e. :

$$\begin{cases} \mathbf{s}_{(0,2,3)}^1 &= \mathbf{s}_{(1,1,3)}^0 + \mathbf{s}_{(1,3,1)}^1 - \mathbf{s}_{(0,4,1)}^1 \\ \mathbf{s}_{(0,3,2)}^1 &= \mathbf{s}_{(3,1,1)}^0 + \mathbf{s}_{(1,1,3)}^1 - \mathbf{s}_{(0,1,4)}^1 \\ \mathbf{s}_{(1,2,2)}^1 &= \frac{1}{2}(\mathbf{s}_{(1,1,3)}^0 + \mathbf{s}_{(1,3,1)}^1 - \mathbf{s}_{(0,4,1)}^1) + \mathbf{s}_{(1,1,3)}^1 - \mathbf{s}_{(0,1,4)}^1 + \mathbf{U}_2 + \tilde{\mathbf{U}}_2. \end{cases} \tag{6.12}$$

En remplaçant ces valeurs dans 6.11, on obtient $\bar{\mathbf{s}}^1 = \mathbf{a}^1 + M^1 \bar{\mathbf{s}}$, avec

$$\mathbf{a}_{(i,j,k)}^1 = \begin{cases} \mathbf{s}_{(1,3,1)}^1 - \mathbf{s}_{(0,4,1)}^1, & \text{si } (i, j, k) = (0, 2, 3), \\ \mathbf{s}_{(1,1,3)}^1 - \mathbf{s}_{(0,1,4)}^1, & \text{si } (i, j, k) = (0, 3, 2), \\ \frac{1}{2}(\mathbf{s}_{(1,1,3)}^1 + \mathbf{s}_{(1,3,1)}^1) - \mathbf{s}_{(0,4,1)}^1 - \mathbf{s}_{(0,1,4)}^1 + \mathbf{U}_2 + \tilde{\mathbf{U}}_2, & \text{si } (i, j, k) = (1, 2, 2), \\ \mathbf{s}_{(i,j,k)}^1, & \text{sinon,} \end{cases} \quad (i, j, k) \in I_5,$$

et

$$\mathbf{M}_{i,j}^1 = \begin{cases} \frac{1}{2}, & \text{si } (i, j) \in \{(13, 1), (13, 2)\}, \\ 1, & \text{si } (i, j) \in \{(14, 1), (17, 2)\}, \quad i = 1, \dots, 21, \quad j = 1, 2, 3. \\ 0, & \text{sinon,} \end{cases}$$

On obtient des formules similaires pour $E_k(S^2)$, $E_k(S^3)$ et $E_k(S^0)$. \square

Lemme 8 *La minimisation de l'énergie 6.10 revient à résoudre le système*

$$\mathbf{L} \bar{\mathbf{s}} = -\mathbf{R}, \quad (6.13)$$

où \mathbf{L} est la matrice 3×3 donnée par

$$\mathbf{L} = \sum_{j=0}^3 [M^j]^T A^k [M^j]$$

et \mathbf{R} est le vecteur 3×3 donné par

$$\mathbf{R} = \sum_{j=0}^3 [M^j]^T A^k \mathbf{a}^j$$

Preuve:

La fonction $\sigma(\bar{\mathbf{s}})$ est quadratique convexe en $\bar{\mathbf{s}}$, son minimum est atteint à la racine de son gradient, i.e. pour $\frac{\partial \sigma}{\partial \bar{\mathbf{s}}} = 0$. $\sigma(\bar{\mathbf{s}})$ est la composée des deux fonctions $\bar{\mathbf{s}} \rightarrow \bar{\mathbf{s}}^T A^k \bar{\mathbf{s}}$ et $\bar{\mathbf{s}} \rightarrow \sum_{j=0}^3 (\mathbf{a}^j + M^j \bar{\mathbf{s}})$. Donc le gradient de $\sigma(\bar{\mathbf{s}})$ est donné par :

$$\frac{\partial \sigma}{\partial \bar{\mathbf{s}}} = 2 \sum_{j=0}^3 [M^j]^T A^k (\mathbf{a}^j + M^j \bar{\mathbf{s}}) = 0,$$

d'où

$$\left(\sum_{j=0}^3 [M^j]^T A^k M^j \right) \bar{\mathbf{s}} = - \sum_{j=0}^3 ([M^j]^T A^k \mathbf{a}^j).$$

\square

La matrice \mathbf{L} du système 6.13 n'est pas singulière. Il existe toujours une solution unique pour les trois points twist par cette méthode. Il faut noter que la matrice \mathbf{L} et sa matrice inverse peuvent être calculées une fois pour toutes avant l'interpolation.

Remarque 12 *Les deux matrices A_2 et A_3 qui contiennent l'intégrale sur les produits des dérivées secondes et tierces des polynômes de Bernstein de degré 5 ont été calculées une fois pour toutes numériquement par des formules de quadrature et stockés comme des constantes dans le programme. Cela évite d'avoir à les évaluer à chaque appel de la routine de minimisation d'énergie sur les facettes.*

6.2.10 Deuxième approche : les six points internes de S^m sont déterminés par la minimisation d'une énergie

La deuxième approche consiste à calculer l'ensemble des six points internes de la facette centrale par minimisation de l'énergie donnée par la formule 6.10. Nous suivons les mêmes procédures que lors de la deuxième étape de l'approche précédente, mais avec $\bar{\mathbf{s}} = \{ \mathbf{s}_{(3,1,1)}^0, \mathbf{s}_{(2,2,1)}^0, \mathbf{s}_{(1,3,1)}^0, \mathbf{s}_{(2,1,2)}^0, \mathbf{s}_{(1,2,2)}^0, \mathbf{s}_{(1,1,3)}^0 \}$ qui contient maintenant tous les six points libres. La fonction à minimiser est donnée par

$$\sigma(\bar{\mathbf{s}}) = \sum_{j=0}^3 [M^j \bar{\mathbf{s}} + \mathbf{a}^j]^T A^k [M^j \bar{\mathbf{s}} + \mathbf{a}^j],$$

avec M^j , $j = 0, 1, 2, 3$ trois matrices 21×6 , \mathbf{a}^j trois vecteurs (21×3) ne dépendant pas des six points de $\bar{\mathbf{s}}$ et A_k la même matrice (21×21) que pour la première approche.

La minimisation de cette fonction nous conduit à résoudre le système linéaire suivant :

$$\mathbf{L}' \bar{\mathbf{s}} = -\mathbf{R}', \quad (6.14)$$

où \mathbf{L}' est la matrice 6×6 donnée par

$$\mathbf{L}' = \sum_{j=0}^3 [M^j]^T A^k [M^j]$$

et \mathbf{R}' est le vecteur 6×3 donné par

$$\mathbf{R}' = \sum_{j=0}^3 [M^j]^T A^k \mathbf{a}^j.$$

Résumé : *Pour la méthode de base du 4-split et pour ces deux généralisations, les six degrés de liberté vectoriels qui sont disponibles pour chaque macro-facette ont été calculés par deux approches différentes. Les deux approches proposées tendent à minimiser une énergie de flexion des facettes.*

6.2.11 Résultats et discussion

L'utilisation d'une minimisation d'énergie s'est avérée être la méthode la plus efficace pour remplir les facettes de manière lisse, une fois les courbes frontière fixées. L'énergie E_3 donne des formes plus arrondies que l'énergie E_2 . Ceci est dû à la nature des vraies intégrales d'énergie dont E_2 et E_3 sont les versions linéarisées. Pour plus de détails à ce sujet, voir les travaux de Moreton, Séquin qui ont étudié ces fonctionnelles et leurs applications pratiques dans [95],[96], [97].

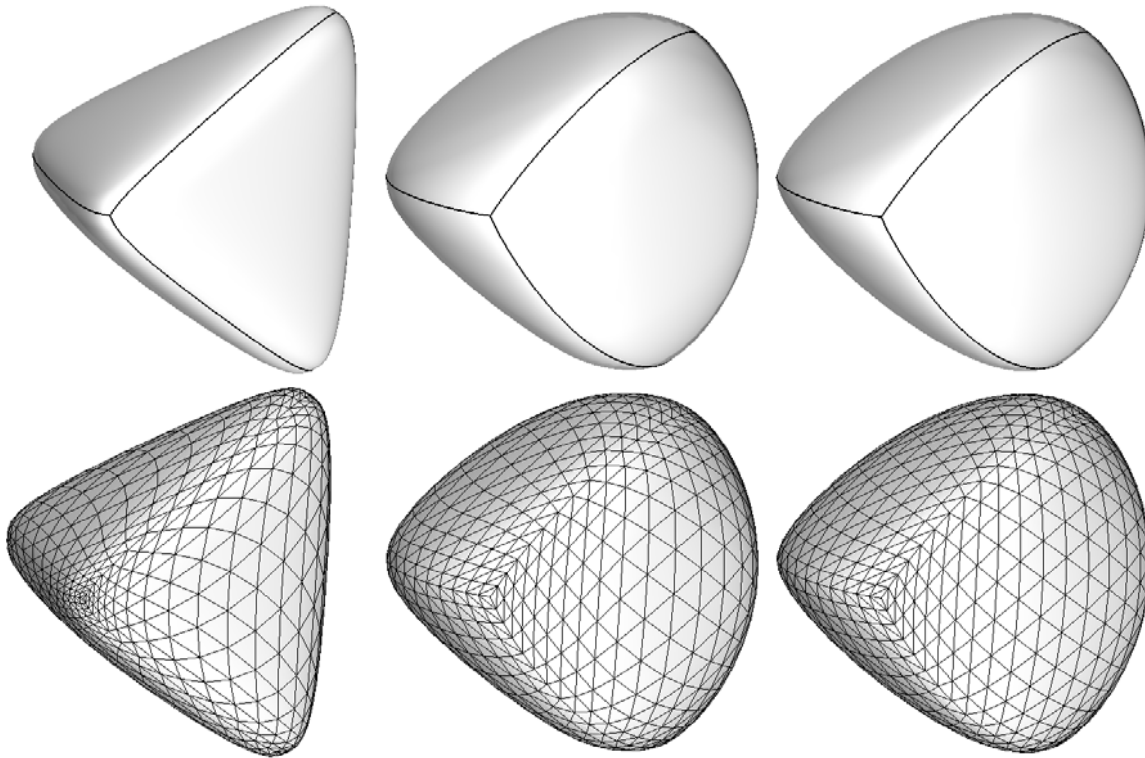


FIG. 6.12 – *Interpolation d'un tétraèdre.*

Pour visualiser les effets des différentes méthodes qui ont été présentées sur le choix des dérivées secondes et le remplissage des facettes, nous montrerons l'exemple du tétraèdre en fig. 6.12. C'est un exemple simple, mais il illustre bien l'influence des paramètres et des points libres. La ligne en haut montre trois surfaces avec les courbes frontière des macro-facettes. La ligne en bas montre en plus des lignes isoparamétriques. Toutes les surfaces interpolent les sommets du tétraèdre.

La surface à gauche correspond à un choix par défaut des paramètres $\beta = 0.15$, $\gamma_1 = 2.0$, $\gamma_2 = 0.0$, ce qui a permis de trouver des courbes lisses "à la main". Ces valeurs sont les mêmes pour tous les sommets à cause de la symétrie du réseau surfacique. Le remplissage

des facettes s'est fait avec des heuristiques simples combinant la normale de la facette sous-jacente et les dérivées transfrontières au milieu des courbes frontière. C'est une des méthodes qui a précédé les techniques présentées ici.

La surface du milieu a des courbes frontière optimisées. Les voisins virtuels optimaux (sect. 6.2.3) ont été calculés, et les paramètres optimaux sont $\gamma_1 = 2.393, \gamma_2 = 0.205$. L'énergie des faces sans optimisation des points internes est égale à 732.

La surface à droite a les mêmes courbes frontières que celle du milieu, mais les points intérieurs des macro-facettes ont été obtenus par minimisation de l'énergie E_3 donnée par la formule 6.9. L'énergie des facettes décroît à 309.

On peut constater que toutes les surfaces sont lisses, avec des courbes frontière un peu plus marquées pour les paramètres manuels. La minimisation d'énergie pour les facettes semble avoir un bon effet supplémentaire sur les lignes isoparamétriques : elles sont distribuées plus régulièrement sur les facettes de la surface à droite.

6.3 Paramètres de forme de la deuxième généralisation

La deuxième généralisation du 4-split (chapitre 5) est beaucoup plus puissante que les autres versions du 4-split, car elle dispose de deux libertés vectorielles pour le choix des dérivées premières et d'un maximum de liberté, i.e. de n ou $n - 1$ (suivant la parité de n) degrés de libertés vectoriels pour le choix des dérivées secondes autour d'un sommet de degré n . Voyons maintenant comment utiliser ces paramètres pour obtenir un meilleur contrôle de la surface. Nous détaillerons dans la sect. 6.3.1 le choix d'une base optimale du plan tangent et dans la sect. 6.3.2 le calcul optimal des dérivées secondes.

6.3.1 Choix de la base du plan tangent

Considérons un sommet quelconque \mathbf{v} de degré n . Dans la section 5.4.2, les vecteurs tangents \mathbf{r}_i^1 autour de \mathbf{v} sont définis en fonction de deux degrés de libertés vectoriels \mathbf{a} et \mathbf{b} , qui constituent une base du plan tangent en \mathbf{v} de la surface, plan contenant tous les vecteurs \mathbf{r}_i^1 autour de \mathbf{v} .

La relation entre \mathbf{r}_i^1 et $\{\mathbf{a}, \mathbf{b}\}$ est donnée par :

$$\mathbf{r}_i^1 = \cos\left(\frac{2\pi(i-1)}{n}\right) \mathbf{a} + \sin\left(\frac{2\pi(i-1)}{n}\right) \mathbf{b}, \quad i = 1, \dots, n. \quad (6.15)$$

Nous pouvons utiliser ces deux vecteurs libres pour interpoler le vecteur normal en \mathbf{v} ou pour approcher, aux sens des moindres carrés, n tangentes $\mathbf{r}_{opt\ i}^1$, définies par une méthode heuristique comme celle de Piper (sect. 6.2.1). Les meilleures valeurs de \mathbf{a} et \mathbf{b} sont déterminées par la solution du problème de minimisation suivant :

$$\|\bar{\mathbf{r}}^1(\mathbf{a}, \mathbf{b}) - \bar{\mathbf{r}}_{opt}^1\|^2 \longrightarrow \min_{\{\mathbf{a}, \mathbf{b}\}}.$$

L'interpolation du vecteur normal en \mathbf{v} s'effectue en choisissant les deux vecteurs \mathbf{a} et \mathbf{b} linéairement indépendants dans le plan passant par \mathbf{v} et orthogonal au vecteur normal. Le degré de liberté restant peut être utilisé pour optimiser la distribution des vecteurs \mathbf{r}_i^1 autour de \mathbf{v} .

6.3.2 Choix optimal des degrés de liberté des dérivées secondes

D'après la section 5.4.4, on peut choisir $\bar{\mathbf{r}}^2$ (vecteur des dérivées secondes des courbes frontière autour de \mathbf{v}) quelconque dans l'espace $\text{Im}(\mathbf{M}_t)$, i.e. on peut prendre un vecteur $(n \times 3)$ $\tilde{\mathbf{r}}^2$ quelconque et le multiplier par la matrice de twist, i.e.

$$\bar{\mathbf{r}}^2 = \mathbf{M}_t \tilde{\mathbf{r}}^2. \quad (6.16)$$

Pour un choix optimal, on calcule d'abord un vecteur $\bar{\mathbf{r}}_{opt}^2$ contenant les dérivées secondes souhaitées. Le choix de ce vecteur dépend beaucoup des applications. Pour un choix automatique nous utilisons le concept du point et de la tangente cibles qui a été introduit dans la section 6.2.6. Ensuite nous déterminons le vecteur $\tilde{\mathbf{r}}^2$ tel que $\bar{\mathbf{r}}^2$ approche au mieux $\bar{\mathbf{r}}_{opt}^2$. On cherche donc $\tilde{\mathbf{r}}^2$, solution du problème de minimisation au sens des moindres carrés suivant :

$$\|\mathbf{M}_t \tilde{\mathbf{r}}^2 - \bar{\mathbf{r}}_{opt}^2\|^2 \longrightarrow \min_{\tilde{\mathbf{r}}^2}.$$

Deux cas peuvent se présenter lors de la solution de ce problème, suivant la parité de n (degré de \mathbf{v})

1. si n est impair : la matrice \mathbf{M}_t est de rang n , et dans ce cas les valeurs optimales des dérivées secondes proposées sont exactement interpolées,
2. si n est pair : la matrice \mathbf{M}_t est seulement de rang $n - 1$ et les dérivées secondes optimales sont assez bien approchées aux moindres carrés. En d'autres termes, le vecteur $\bar{\mathbf{r}}^2 = \mathbf{M}_t \tilde{\mathbf{r}}^2$ est très proche de $\bar{\mathbf{r}}_{opt}^2$, car la matrice \mathbf{M}_t est presque de rang complet.

Pour les deux cas, nous résolvant en $\tilde{\mathbf{r}}^2$ le système linéaire

$$\mathbf{M}_t^T \mathbf{M}_t \tilde{\mathbf{r}}^2 = \mathbf{M}_t^T \bar{\mathbf{r}}_{opt}^2,$$

avec une méthode numérique (tirée de la librairie LAPACK) permettant de trouver toujours une solution de ce système.

Une fois $\tilde{\mathbf{r}}^2$ calculé, les dérivées secondes $\bar{\mathbf{r}}^2$ sont obtenues de 6.16, i.e.

$$\begin{cases} \bar{\mathbf{r}}_1^2 = \frac{1}{2} \tilde{\mathbf{r}}_n^2 + \frac{1}{2} \tilde{\mathbf{r}}_1^2, \\ \bar{\mathbf{r}}_i^2 = \frac{1}{2} \tilde{\mathbf{r}}_{i-1}^2 + \frac{1}{2} \tilde{\mathbf{r}}_i^2, \quad i = 2, \dots, n. \end{cases}$$

Résumé: *Les deux degrés de liberté pour les dérivées premières des courbes frontière sont utilisés dans l'approximation des points de Piper avec la possibilité d'interpoler un vecteur normal donné au sommet. Pour les dérivées secondes les techniques du point et de la tangente cible ont été adaptées.*

6.3.3 Résultats et discussion

L'étude approfondie des matrices de la continuité G^1 en un sommet du réseau a mené à l'introduction de degrés de liberté supplémentaires, notamment pour les dérivées secondes des courbes frontière. Les courbes peuvent ainsi être manipulées plus librement dans le but d'obtenir une forme optimale. Dans les sections précédentes nous avons tenté de répondre aux questions suivantes : Quelle est la forme optimale des courbes frontière et de la surface, et comment les atteindre ? Les seules données concrètes dont on dispose sont les points à interpoler et la triangulation sous-jacente que la surface doit "habiller". Une infinité de solutions est possible avec ce schéma d'interpolation. Pour pouvoir obtenir de manière automatique une surface interpolante par défaut qui soit globalement satisfaisante, tous les degrés de libertés doivent être fixés. Pour cela il n'est possible que de se baser sur les données concrètes d'entrée. Dans cette optique nous avons développé des règles heuristiques qui tiennent compte de la géométrie du réseau en calculant des points et des tangentes cibles par une règle de subdivision.

Exemple 1 et 2: Un premier exemple montre une sorte de valise, i.e. un cube avec des coins arrondis, fig. 6.13. L'autre exemple est un cube déformé. Deux sommets du cubes original ont été modifiés, voir à gauche de fig. 6.14. Les deux surfaces interpolantes qui sont montrées au milieu de la figure correspondent à un choix des points et de la tangente cible par défaut par subdivision. Ces résultats sont très satisfaisants, car les surfaces sont suffisamment lisses. Ces exemples ont l'air simples et faciles. Ce sont des exemples académiques qui ont l'air triviaux. Pourtant ils ont été choisis pour le degré pair de tous les sommets du réseau. Cela veut dire que c'est la situation la moins favorable qui puisse arriver lors du choix des dérivées secondes. Rappelons que dans ce cas les dérivées dites optimales ne peuvent qu'être approchées et non interpolées. On observe également que le choix des points et tangentes cibles préserve la symétrie du réseau.

Le coloriage des facettes du polygone de contrôle à droite dans les fig. 6.13, 6.14 permet de distinguer chacune des quatre facettes de Bézier quintiques, composant les macro-facettes.

Exemple 3: Un exemple complexe est montré en fig. 6.15. Ce réseau surfacique est complexe dans plusieurs sens. Il représente

- un objet de topologie complexe

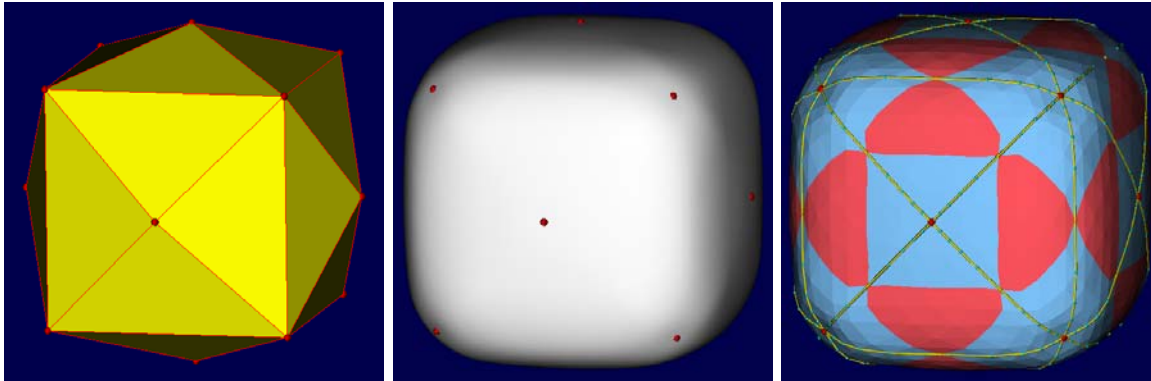


FIG. 6.13 – Réseau surfacique – surface lisse – polygone de contrôle.

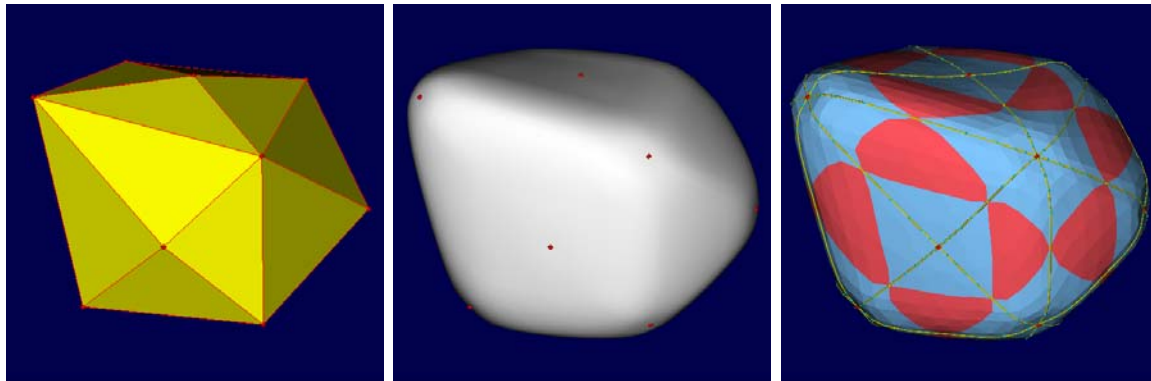


FIG. 6.14 – Réseau surfacique – surface lisse – polygone de contrôle.

- ayant plusieurs trous,
- ayant des bords,
- ayant une géométrie interne avec des courbure de Gauss positive, nulles et négatives,
et
- des sommets de degrés très divers, parfois même très élevés.

La fig. 6.15 montre en haut le réseau surfacique et la surface interpolante avec les courbes frontière. En bas un zoom est effectué sur un sommet de degré 8. Les bords n'ont pas encore été traité dans cet exemple, il le seront en sect. 6.4.

Exemple 4: Maintenant l'exemple qui est le plus important à nos yeux, celui du tore. Le tore est une sorte de benchmark (test de référence) pour les méthodes d'interpolation de maillages arbitraires. Le tore a une géométrie complexe et possède de plus dans notre cas uniquement des sommets de degré 6, qui est un nombre pair. Dans l'état de l'art de Mann et al. [92] le tore a été l'exemple sur lequel toutes les méthodes existantes ont été testées.

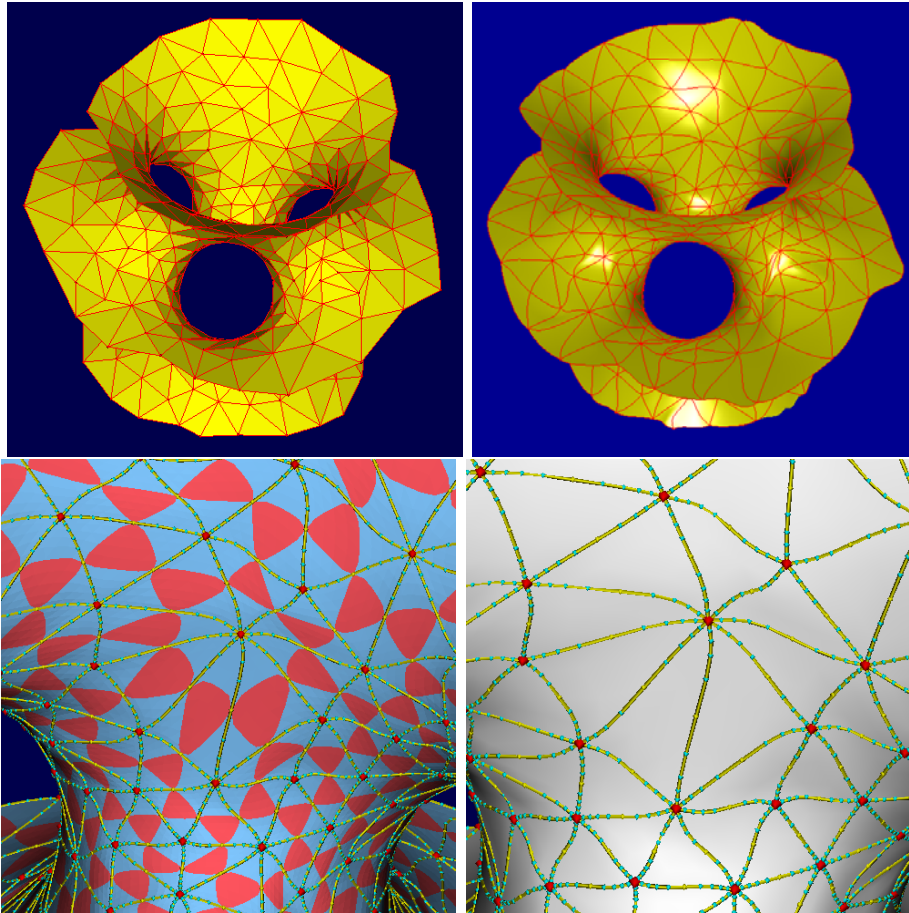


FIG. 6.15 – *En haut : polygone de contrôle – surface avec courbes frontière. En bas : zoom sur un sommet de degré 8.*

Grâce au choix libre des dérivées secondes nous sommes également en mesure d’interpoler le maillage du tore par une surface lisse, voir fig. 6.16.

Exemple 5 : Nous utilisons maintenant le maillage du tore pour illustrer la sensibilité du schéma d’interpolation par rapport au choix des degrés de liberté. En fig. 6.17 on voit deux surfaces interpolant le même réseau du tore (fig. 6.16 à gauche). A gauche la surface qu’on qualifiera généralement comme “correcte”, car elle reproduit la forme du réseau en l’habillant de manière lisse.

A droite seules les tangentes cible pour le calcul des dérivées secondes ont été modifiées. Le résultat est une sorte de “doughnut”, qui a encore la topologie du tore, mais exhibe de fortes ondulations. Une telle forme peut naturellement être souhaitée.

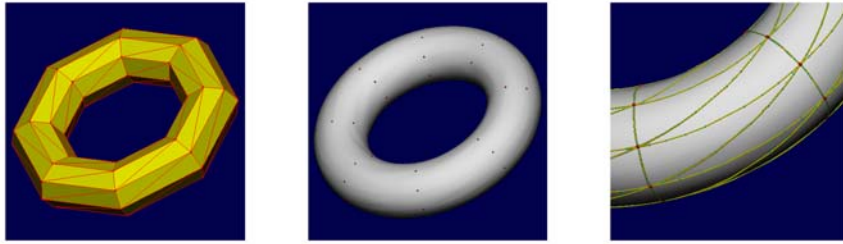


FIG. 6.16 – Maillage d'un tore – surface – zoom sur les courbes frontière.

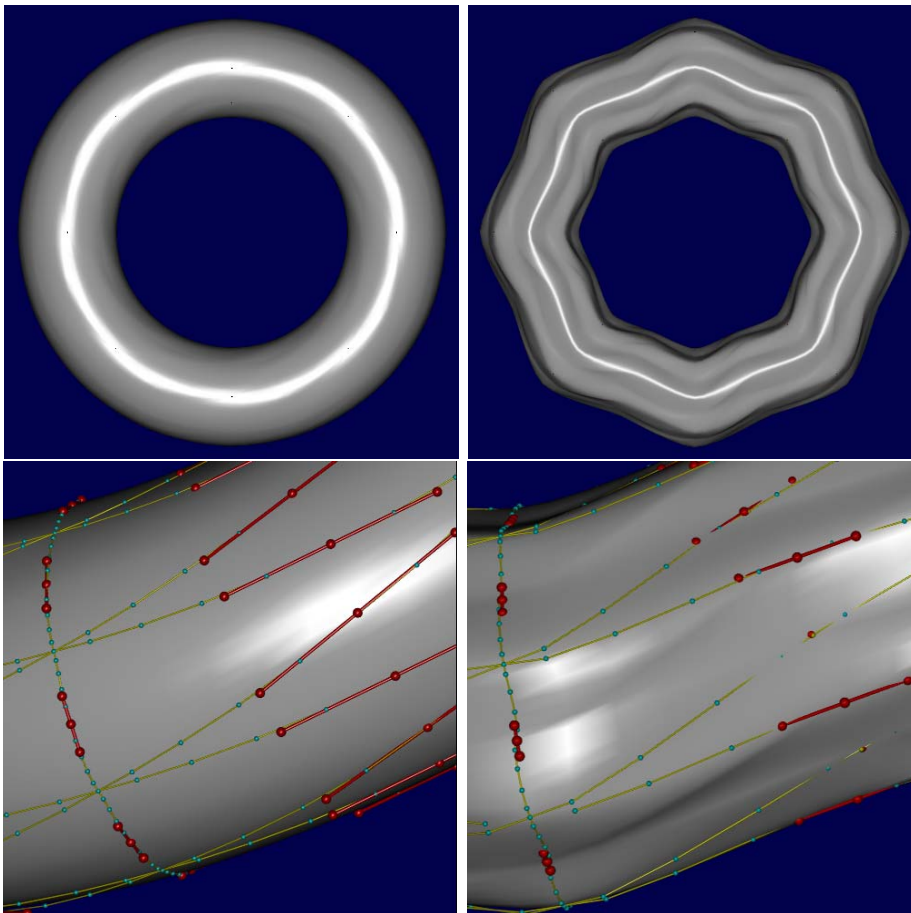


FIG. 6.17 – interpolation d'un maillage torique; à droite avec de mauvais points cibles, à gauche avec des points cibles corrects.

6.4 Traitement des réseaux surfaciques ouverts

Dans cette section, nous allons exposer une approche spéciale pour traiter les réseaux surfaciques avec bord. Une arête du réseau est dite de bord si elle n'appartient qu'à un seul triangle. Un sommet appartient au bord du réseau si deux arêtes en ce sommet appartiennent au bord. Nous allons utiliser la stricte localité des deux méthodes exposées dans les chapitres 4 et 5 pour traiter d'une manière raisonnable les courbes du bord et leurs tangentes trans-frontière. Plus précisément, nous allons proposer une méthode qui assure que les courbes de bord sont de continuité G^1 .

6.4.1 Description de l'algorithme : traitement du bord

Les méthodes du chap. 4 et 5 sont strictement locales. Elles déterminent les points de contrôle de la surface en trois étapes :

- pour chaque sommet : les points de contrôle des courbes frontières incidentes au sommet.
- pour chaque arête : les points de la première rangée intérieure des facettes incidentes au sommet.
- pour chaque facette : les 15 points intérieurs libres.

Seules les deux premières étapes sont à revoir pour un traitement de réseaux ouverts, i.e. de réseaux avec des bords. L'étape (1.) consiste à déterminer les dérivées premières et secondes $\bar{\mathbf{r}}^1, \bar{\mathbf{r}}^2$ des courbes en les sommets. Tous les autres points de contrôle des courbes sont ainsi déterminés.

Considérons un sommet \mathbf{v} de degré n sur le bord du réseau surfacique. Soit $\bar{\mathbf{p}} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ le voisinage réel \mathbf{v} (i.e. l'ensemble des sommets directement voisins).

La construction des macro-facettes correspondant aux triangles du bord (cf. fig 6.18) passe par les étapes suivantes :

- (1) compléter, de façon symétrique, le voisinage de \mathbf{v} afin d'obtenir une distribution uniforme des vecteurs tangents, donnés par la formule 6.17,
- (2) calculer les dérivées premières et secondes $\bar{\mathbf{r}}^1$ et $\bar{\mathbf{r}}^2$ et les dérivées trans-frontière autour de \mathbf{v} , en utilisant le nouveau voisinage virtuel, calculé dans l'étape précédente, comme si \mathbf{v} était un sommet intérieur,
- (3) enlever les facettes rajoutées par l'introduction des sommets de la première étape.

Le calcul de $\bar{\mathbf{r}}^1, \bar{\mathbf{r}}^2$ et des tangentes trans-frontière peut être effectué suivant l'une des deux méthodes présentées dans les chapitres 4 et 5. Comme nous avons vu, ces deux méthodes utilisent le voisinage du sommet. Ici, la méthode utilise le voisinage complété

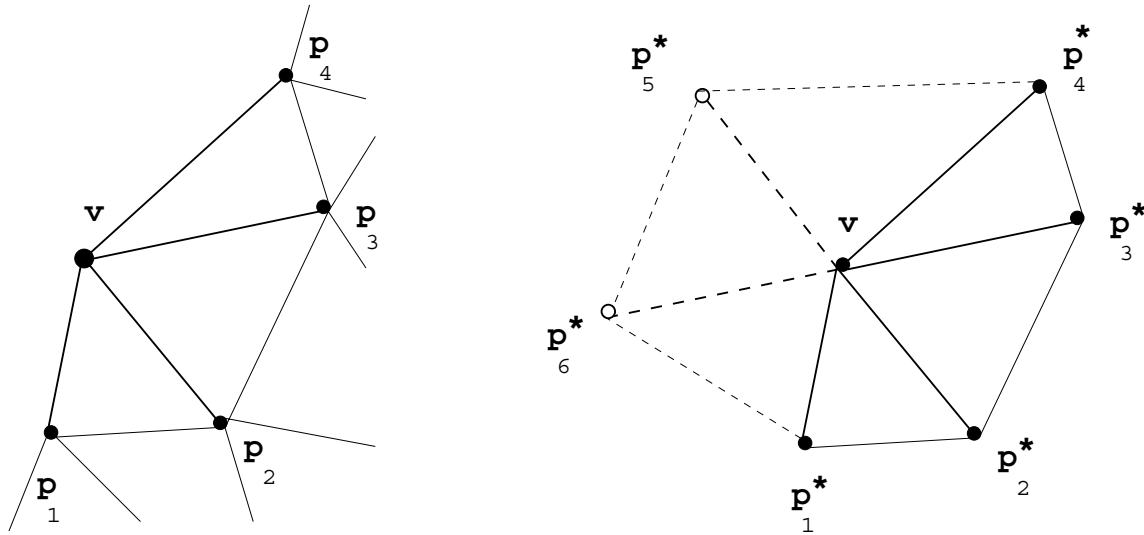


FIG. 6.18 – *Le voisinage réel d'un sommet du bord de degré 4 et son voisinage complété.*

avec des points supplémentaires. Nous pouvons déjà faire les remarques suivantes sur cet algorithme :

Remarque 13 *Le complément du voisinage de v est indispensable, car la construction de \bar{r}^1 et \bar{r}^2 suppose que le sommet est à l'intérieur du réseau (cf. l'interprétation géométrique donnée à la construction de \bar{r}^1 dans la section 4.3.2). Mais un sommet sur le bord peut être supposé à l'intérieur en imaginant qu'il est entouré par des sommets virtuels (calculés symétriquement à partir du voisinage réel); après avoir effectué les calculs, ces sommets seront supprimés, et on ne garde que les facettes correspondant aux triangles du réseau.*

Remarque 14 *Le calcul des points de contrôle intérieurs libres des facettes du bord s'effectue, localement, avec les procédures expliquées dans la section 6.2.8.*

Remarque 15 *Les deux méthodes sont strictement locales et produisent une surface G^1 , donc lorsqu'on enlève les facettes supplémentaires, dans la troisième étape, les facettes qui restent se raccordent toujours avec la continuité G^1 .*

6.4.2 Complément du voisinage d'un sommet au bord du réseau

Le point essentiel de l'algorithme traitant le bord du réseau est celui consistant à compléter le voisinage de chacun de ses sommets, de telle manière que les courbes de bord soient G^1 . Cet algorithme complète le voisinage de v en passant par les étapes suivantes :

- calculer le nouveau nombre des points du voisinage virtuel par

$$N = 2n - 2.$$

- si $n = 2$, la méthode ne calcule aucun point symétrique, car les deux points de $\bar{\mathbf{p}}$ sont sur le bord, et aucun joint G^1 n'est à calculer. Les deux morceaux de courbe frontière et leurs tangentes trans-frontière appartenant à ce sommet peuvent être choisis librement.
- si $n > 2$, alors le nouveau voisinage est constitué des $N = 2n - 2$ points donnés par

$$\mathbf{p}_i^* = \begin{cases} \mathbf{p}_i, & i = 1, \dots, n, \\ \mathbf{v} + (\mathbf{v} - \mathbf{p}_{i-n+1}), & i = n + 1, \dots, 2n - 2. \end{cases} \quad (6.17)$$

La figure 6.18 montre cette procédure de complément pour $n = 4$.

Remarque 16 La méthode du chapitre 4 calcule les vecteurs $\bar{\mathbf{r}}^1$, $\bar{\mathbf{r}}^2$ et $\bar{\mathbf{t}}$ à partir des formules 4.8 (sect. 4.1) avec le degré virtuel de $2n - 2$ au lieu de n et le voisinage virtuel ci-dessus. Cependant, la méthode du chapitre 5 utilise ce voisinage pour trouver les degrés de liberté $\bar{\mathbf{a}}$ et $\bar{\mathbf{b}}$ dans la détermination de $\bar{\mathbf{r}}^1$ et pour trouver les N ou les $N - 1$ degrés de liberté dans la détermination de $\bar{\mathbf{r}}^2$. Les méthode de détermination de ces degrés de liberté sont expliqués dans la section 6.3. $\bar{\mathbf{r}}^1$, $\bar{\mathbf{r}}^2$ et $\bar{\mathbf{t}}$ sont respectivement donnés, cette fois, par les formules 6.15, 6.16 et 5.7.

Exemple

L'exemple du hypersheet a été maintenant recalculé. Les bords sont traités avec la méthode décrite plus haut, voir fig. 6.19.

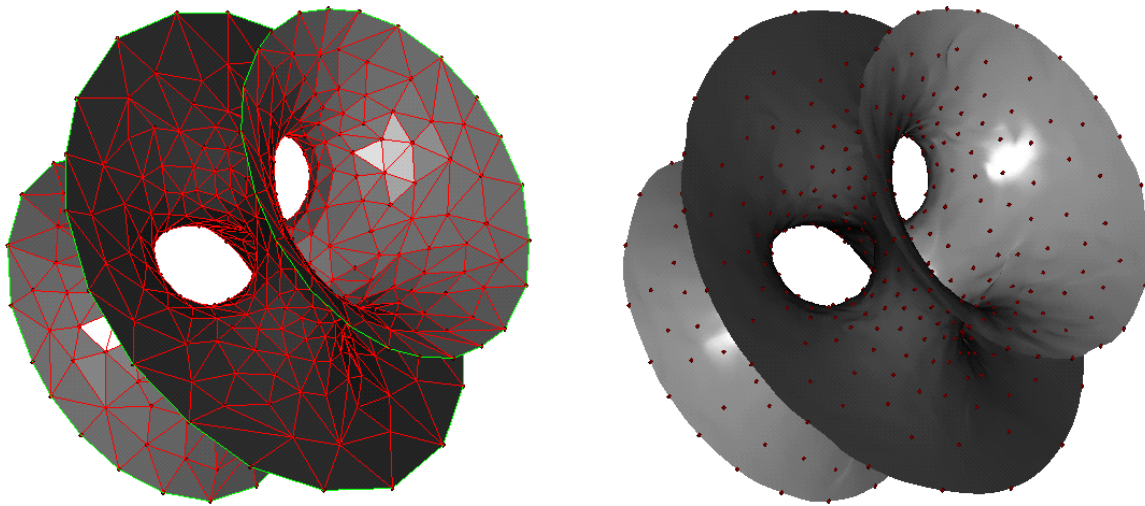


FIG. 6.19 – hypersheet avec courbes de bord.

6.5 Conclusion

Plusieurs algorithmes efficaces ont été développés pour satisfaire les contraintes de forme des surfaces de topologie arbitraire. La majorité de ces algorithmes influence localement la surface, produisant ainsi un schéma local. Un outil de modélisation basé sur la notion de point cible et de tangente cible est mis à disposition de l'utilisateur pour choisir les courbes frontière qui correspondent au mieux avec son application. Nous avons aussi présenté deux approches différentes pour le choix automatique des six points libres de la sous-facette centrale. Enfin, la localité de la méthode a permis de traiter les réseaux surfaciques ouverts. Un exemple a été fourni afin de montrer l'efficacité de cet algorithme.

Nous pensons avoir présenté l'essentiel concernant le choix des paramètres libres des méthodes des chapitres 6 et 5, mais nous sommes très loin d'avoir étudié toute la richesse du schéma. D'autres applications seraient souhaitables. La préservation de forme, l'interpolation ou l'approximation des courbures surfaciques pourraient par exemple être obtenues grâce à une bonne détermination des paramètres libres.

Conclusion

L'objectif de la thèse était d'établir quelques méthodes pour la construction de surfaces de topologie arbitraire, la modélisation de leur forme et de permettre de satisfaire plusieurs propriétés géométriques comme l'interpolation des positions et des normales, la convexité, l'interpolation ou l'approximation de courbures surfaciques, etc ...

Y sommes-nous parvenus ?

Nous savions depuis le début qu'un tel objectif ne serait atteint que par une méthode souple et offrant beaucoup de paramètres libres, en relation simple avec les caractéristiques géométriques intrinsèques des surfaces.

Puis, compte tenu de l'importance des surfaces paramétriques polynomiales par morceaux dans beaucoup d'applications fondamentales et plus particulièrement dans la conception géométrique assistée par ordinateur (CGAO), nous avons d'abord étudié les courbes et surfaces de Bézier : courbes d'interpolation, B-splines, splines rationnelles, splines rationnelles non uniformes (NURBS), toutes modélisent intuitivement la forme de la surface à partir d'un petit nombre de points de contrôle et constituent le standard des logiciels de CAO/CFAO. Enfin étant donné que la continuité paramétrique entre les patches dépend de la paramétrisation (et pose des contraintes sur la topologie des réseaux surfaciques) nous avons alors étudié la continuité géométrique, qui est la continuité naturelle entre les patches car elle ne dépend pas des paramétrisations des patches.

Confrontées au problème de twist, les surfaces polynomiales par morceaux imposent quelques contraintes supplémentaires sur la construction des patches. Nous avons donc étudié, dans le chapitre 2, plusieurs techniques de résolution de ce problème. Nous avons vu qu'au fil des années, plusieurs approches ont été proposées pour résoudre ce problème. Mais à cause de la topologie arbitraire des réseaux surfaciques d'entrée, ces méthodes souffrent de certains défauts de forme, comme par exemple la présence d'oscillations, de bosses, de coins, d'auto-intersections ou de non conservation de la forme. Dans un premier temps, nous avons étudié plusieurs approches de la littérature traitant le problème de l'interpolation G^1 d'un réseau surfacique de topologie arbitraire. Puis, en fixant quelques critères de comparaison, nous avons préféré la méthode du 4-split. Le chapitre 3 a été entièrement consacré à une étude détaillée de cette méthode car elle a été choisie comme

une base de toutes les contributions de la thèse. Nous avons été en mesure de montrer quelques propriétés théoriques de cette méthode contribuant ainsi à la version originale de la méthode et présentant aussi une introduction adaptée aux contributions proposées dans les chapitres suivants.

Le chapitre 4 fut consacré à une première généralisation de la méthode de base du 4-split. En s'appuyant sur la remarque que le voisinage d'un sommet, intervenant dans tous les calculs autour de ce sommet, peut être remplacé par un autre ensemble de n points, nous avons alors réussi à localiser la méthode du 4-split et à produire plus de degré de libertés dans le schéma d'interpolation. De plus, l'interpolation des vecteurs normaux est devenue possible, ce qui a offert un outil supplémentaire de design.

Dans le chapitre 5, nous avons revu la technique de résolution du problème de twist en faisant appel à la notion des matrices circulantes. Le but était d'exploiter pleinement la structure de la méthode du 4-split. Le problème de twist a été résolu avec une méthode plus souple et plus général, offrant un nombre maximum de degrés de libertés dans le choix des courbes frontière. Ces nouveaux paramètres sont géométriquement plus significatifs et ont permis de satisfaire beaucoup plus de contraintes de forme.

Le chapitre 6 fut la contribution principale de cette thèse. Il a été consacré au choix des paramètres de forme et à l'exploitation de leurs interprétations géométriques pour satisfaire localement les propriétés de lissage de la surface. La méthode offre des paramètres de forme pour le choix des courbes frontière mais laisse aussi six points de contrôle libres au sein de chaque macro-facettes. Les paramètres des courbes frontières peuvent être choisis soit manuellement en vue d'une modélisation interactive, soit automatiquement sous un certain critère de lissage. Dans le cas interactif, les calculs se font en temps réel, grâce à la stricte localité des algorithmes. Dans le cas du design automatique, de nombreux algorithmes efficaces ont été développés pour satisfaire un certain nombre de caractéristiques de forme. La majorité de ces algorithmes sont strictement locaux. Une nouvelle notion, portant le nom de point cible et tangente cible, a été introduite et qui offre à l'utilisateur un outil efficace pour le choix des courbes frontière. Nous avons aussi présenté deux approches différentes pour le choix automatique des six points libres de la sous-facette centrale. La première approche mélangeait une méthode heuristique et une méthode de minimisation d'énergie. La deuxième approche calculait les six points par une seule minimisation d'énergie.

Nous sommes très loin d'avoir abordé, dans le chapitre 6, toute la richesse du schéma d'interpolation pour le contrôle de la forme: il existe d'autres manières de satisfaire d'autres caractéristiques de forme et des applications telles que la préservation de forme

qui peuvent être obtenues avec une bonne détermination des paramètres disponibles. Une propriété importante du schéma du 4-split, qui le distingue des autres schémas existant, est justement l'idée de couper, en quatre et de façon régulière, les triangles de base. Le découpage fait penser aux travaux sur la multirésolution, que ce soit sur les triangulations sphériques [119, 25, 53, 54, 88, 127] ou sur des surfaces de topologie arbitraire représentées par une triangulation [35, 36]. La connectivité par subdivision uniforme de la triangulation est à la base d'une analyse multirésolution dans ces cas. Cette connectivité de la triangulation devra permettre également l'édition du réseau à différents niveaux de détail d'une manière hiérarchique. Un des travaux futurs sera donc de rendre la méthode hiérarchisable. On pourra insérer un nouveau point de contrôle au milieu d'une arête, en créant alors, dans chaque patch adjacent à cette arête, quatre nouveaux patches "fils". La méthode pourra être répétée plusieurs fois afin de raffiner, de plus en plus, une zone donnée. L'objectif est de pouvoir éditer les sommets que l'on a insérés et donc éditer la surface à plusieurs niveaux de détail, grâce à la localité du schéma d'interpolation. Durant son stage de D.E.A, E. Guilbert [61] a essayé de hiérarchiser cette méthode mais a découvert que cela n'est pas possible dans l'état actuel. Le problème provenait de la définition par morceaux de la fonction scalaire Φ . Si elle était linéaire, le schéma hiérarchique aurait été possible. Une piste qui pourrait servir serait d'essayer de traiter le problème de twist de manière encore plus générale, par exemple en soulevant le choix de $\mu = \nu = \frac{1}{2}$ fait au départ.

Annexe A

Calcul des deux premières rangés des points de contrôle d'un patch

Rappelons qu'une facette triangulaire de Bézier de degré d est définie par

$$B(\lambda_1, \lambda_2, \lambda_3) = \sum_{\substack{i+j+l=d \\ i,j,l \geq 0}} \mathbf{b}_{i,j,l} B_{i,j,l}^d(\lambda_1, \lambda_2, \lambda_3), \quad (\lambda_1, \lambda_2, \lambda_3) \in \Delta,$$

où $\Delta = \{\lambda_1, \lambda_2, \lambda_3 \in [0, 1], \lambda_1 + \lambda_2 + \lambda_3 = 1\}$, $\mathbf{b}_{i,j,k}$ sont les points de contrôle de Bézier de la facette et $B_{i,j,l}^d(\lambda_1, \lambda_2, \lambda_3) = \frac{d!}{i!j!l!} \lambda^i \lambda^j \lambda^l$ sont les polynômes de Bernstein généralisés. Pour plus de détails sur les facettes de Bézier triangulaires voir sect. 1.2 et [43].

Les 4 facettes triangulaires de Bézier de degré 5 qui composent la macro-facette M^i sont notées S^1, S^2, S^3, S^m . Elles sont paramétrées sur les 4 sous-triangles de Δ comme indiqué dans la fig. A.1. Les points de contrôle de Bézier de M sont notés resp. par $\mathbf{s}_{i,j,l}^1, \mathbf{s}_{i,j,l}^2, \mathbf{s}_{i,j,l}^3, \mathbf{s}_{i,j,l}^m$, avec $i + j + l = 5$:

On note aussi $a_k, k = 1, 2, 3$ les sommets de Δ , et $\mathbf{u}_k = a_{k+1} - a_k, k = 1, 2, 3$ les vecteurs entre les sommets de Δ , définissant les directions de dérivation de M . Les trois arêtes du triangle domaine sont notées par $E_k(u) = \mathbf{a}_k(1 - u) + \mathbf{a}_{k+1}u, k = 1, 2, 3, u \in [0, 1]$.

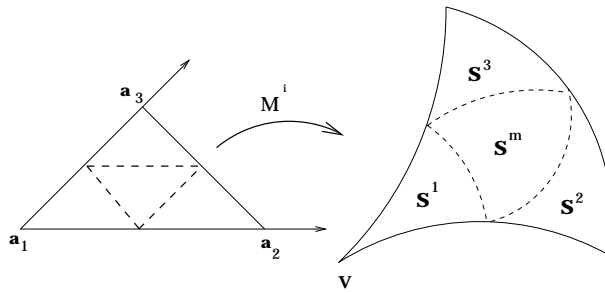


FIG. A.1 – Paramétrisation de la macro-facette M^i .

Les degrés des trois sommets de la macro-facette sont notés n_k , $k = 1, 2, 3$.

A.1 Les points de contrôle de la première (rangée) dérivée de M

Soient \mathbf{b}_i^L , \mathbf{b}_i^R , $i = 0, \dots, 3$ les points de contrôle des deux morceaux de la courbe frontière calculée dans la sect. 3.7, i.e. :

$$M(E_k(u)) = \begin{cases} \sum_{i=0}^3 \mathbf{b}_i^L B_i^3(2u), & u \in [0, \frac{1}{2}] \\ \sum_{i=0}^3 \mathbf{b}_i^R B_i^3(2u-1), & u \in [\frac{1}{2}, 1] \end{cases}, \quad k = 1, 2, 3. \quad (\text{A.1})$$

Les tangentes trans-frontière $[D_{\mathbf{u}_{i+1}} M^i](u_i, 0)$ sont données ici par

$$[-D_{\mathbf{u}_{k+2}} M](E_k(u)) = \Phi_k(u) [D_{\mathbf{u}_k} M](E_k(u)) + \Psi_k(u) \mathbf{V}_k(u), \quad (\text{A.2})$$

où

$$[D_{\mathbf{u}_k} M](E_k(u)) = \begin{cases} 6 \sum_{i=0}^2 (\mathbf{b}_{i+1}^L - \mathbf{b}_i^L) B_i^2(2u), & u \in [0, \frac{1}{2}] \\ 6 \sum_{i=0}^2 (\mathbf{b}_{i+1}^R - \mathbf{b}_i^R) B_i^2(2u-1), & u \in [\frac{1}{2}, 1] \end{cases}, \quad k = 1, 2, 3, \quad (\text{A.3})$$

est la dérivée de la courbe frontière $M(E_k(u))$,

$$\mathbf{V}_k(E_k(u)) = \begin{cases} \sum_{i=0}^2 \mathbf{v}_i^L B_i^2(2u), & u \in [0, \frac{1}{2}] \\ \sum_{i=0}^2 \mathbf{v}_i^R B_i^2(2u-1), & u \in [\frac{1}{2}, 1] \end{cases}, \quad k = 1, 2, 3, \quad (\text{A.4})$$

est la fonction vectorielle de la section 3.9, et Φ_k, Ψ_k sont les fonctions réelles définies dans les sections 3.6 et 3.9 par :

$$\Phi_k(u) = \begin{cases} (1-2u) \cos \frac{2\pi}{n_k} + u, & u \in [0, \frac{1}{2}] \\ (1-u) + (2u-1)(1 - \cos \frac{2\pi}{n_{k+1}}), & u \in [\frac{1}{2}, 1], \end{cases} \quad (\text{A.5})$$

et

$$\Psi_k(u) = (1-u) \sin \frac{2\pi}{n_k} + u \sin \frac{2\pi}{n_{k+1}}, \quad u \in [0, 1]. \quad (\text{A.6})$$

Considérons maintenant la frontière de M correspondant à E_1 partagée par les sous-facettes S^1 et S^2 . Les points de contrôle sont indexés comme dans la fig. A.2

Puisque nous construisons des sous-facettes quintiques, la courbe frontière donnée par A.1 pour $k = 1$ doit être élevée du degré 3 au degré 5, ce qui donne les points de contrôle

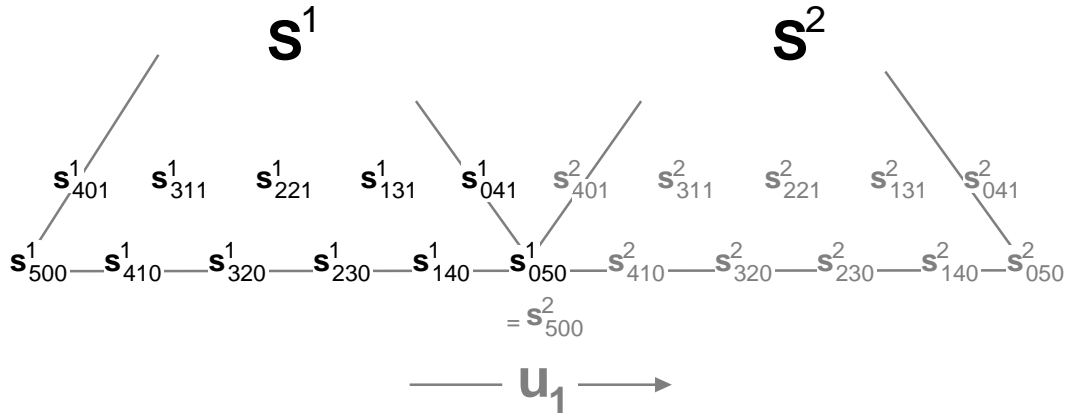


FIG. A.2 – points de contrôle de la courbe frontière et de la première rangée intérieure de M correspondant au bord \mathbf{u}_1 .

suyvants de la courbe frontière de M correspondant à l'arête $E_1(u)$:

$$\begin{aligned}
\mathbf{s}_{(5,0,0)}^1 &= \mathbf{b}_0^L \\
\mathbf{s}_{(4,1,0)}^1 &= \frac{2}{5}\mathbf{b}_0^L + \frac{3}{5}\mathbf{b}_1^L \\
\mathbf{s}_{(3,2,0)}^1 &= \frac{1}{10}\mathbf{b}_0^L + \frac{3}{5}\mathbf{b}_1^L + \frac{3}{10}\mathbf{b}_2^L \\
\mathbf{s}_{(2,3,0)}^1 &= \frac{3}{10}\mathbf{b}_1^L + \frac{3}{5}\mathbf{b}_2^L + \frac{1}{10}\mathbf{b}_3^L \\
\mathbf{s}_{(1,4,0)}^1 &= \frac{3}{5}\mathbf{b}_2^L + \frac{2}{5}\mathbf{b}_3^L \\
\mathbf{s}_{(0,5,0)}^1 &= \mathbf{b}_3^L.
\end{aligned} \tag{A.7}$$

Les points $\mathbf{s}_{5-i,i,0}^2$, $i = 0, \dots, 5$ sont trouvés de la même manière à partir des points \mathbf{b}_i^R , $i = 0, \dots, 3$.

La première rangée des points de contrôle intérieurs $\mathbf{s}_{4-i,i,0}^1$ et $\mathbf{s}_{4-i,i,0}^2$, $i = 0, \dots, 4$ se détermine grâce aux dérivées trans-frontière premières :

$$[-D_{\mathbf{u}_3}M](E_1(u)) = \begin{cases} 10 \sum_{i=0}^4 (\mathbf{s}_{(4-i,i,1)}^1 - \mathbf{s}_{(5-i,i,0)}^1) B_i^4(2u), & u \in [0, \frac{1}{2}] \\ 10 \sum_{i=0}^4 (\mathbf{s}_{(4-i,i,1)}^2 - \mathbf{s}_{(5-i,i,0)}^2) B_i^4(2u-1), & u \in [\frac{1}{2}, 1] \end{cases} \tag{A.8}$$

Mais d'après A.2

$$[-D_{\mathbf{u}_3}M](E_1(u)) = \Phi_1(u) [D_{\mathbf{u}_1}M](E_1(u)) + \Psi_1(u) \mathbf{V}_1(u). \tag{A.9}$$

En remplaçant $\Phi_1(u)$, $[D_{\mathbf{u}_1}M](E_1(u))$, $\Psi_1(u)$ et $\mathbf{V}_1(u)$ par leurs valeurs prises respectivement des équations A.5, A.3, A.6 et A.4 avec $k = 1$, en faisant les produits $\Phi_1(u) [D_{\mathbf{u}_1}M](E_1(u))$

et $\Psi_1(u)\mathbf{V}_1(u)$, et en utilisant la règle suivante de la multiplication des polynômes de Bernstein :

$$B_i^p(u)B_j^q(u) = \frac{C_i^p C_j^q}{C_{p+q}^{i+j}} B_{i+j}^{p+q}(u), \quad (\text{A.10})$$

l'équation A.9 devient alors :

$$[-D_{\mathbf{u}_3} M](E_1(u)) = \begin{cases} \sum_{i=0}^3 \left[\left(1 - \frac{i}{3}\right) (6c_1(\mathbf{b}_{i+1}^L - \mathbf{b}_i^L) - s_1 \mathbf{v}_i^L) \right. \\ \quad \left. + \frac{i}{3} (3(\mathbf{b}_i^L - \mathbf{b}_{i-1}^L) - s_2 \mathbf{v}_{i-1}^L) \right] B_i^3(2u), & u \in [0, \frac{1}{2}] \\ \sum_{i=0}^3 \left[\left(1 - \frac{i}{3}\right) (3(\mathbf{b}_{i+1}^R - \mathbf{b}_i^R) - \frac{1}{2}(s_1 + s_2) \mathbf{v}_i^R) \right. \\ \quad \left. + \frac{i}{3} (6c_2(\mathbf{b}_i^R - \mathbf{b}_{i-1}^R) - s_2 \mathbf{v}_{i-1}^R) \right] B_i^3(2u - 1), & u \in [\frac{1}{2}, 1]. \end{cases} \quad (\text{A.11})$$

avec $c_j = \cos(\frac{2\pi}{n_j})$ et $s_j = \sin(\frac{2\pi}{n_j})$.

Élevons au degré quatre chaque morceau cubique de la dérivée trans-frontière A.11 puis comparons celle-ci avec A.2, on obtient alors les points de contrôle de la première rangée intérieure de M correspondant à l'arête $E_1(u)$:

$$\begin{aligned} \mathbf{s}_{(4,0,1)}^1 &= \mathbf{s}_{(5,0,0)}^1 - \frac{3c_1}{5} \mathbf{b}_0^L + \frac{3c_1}{5} \mathbf{b}_1^L + \frac{s_1}{10} \mathbf{v}_0^L \\ \mathbf{s}_{(3,1,1)}^1 &= \mathbf{s}_{(4,1,0)}^1 - \frac{3}{40} (2c_1 + 1) \mathbf{b}_0^L + \frac{3}{40} (1 - 2c_1) \mathbf{b}_1^L + \frac{3c_1}{10} \mathbf{b}_2^L + \frac{1}{80} (3s_1 + s_2) \mathbf{v}_0^L + \frac{s_1}{20} \mathbf{v}_1^L \\ \mathbf{s}_{(2,2,1)}^1 &= \mathbf{s}_{(3,2,0)}^1 - \frac{1}{20} \mathbf{b}_0^L - \frac{1}{20} (1 + 4c_1) \mathbf{b}_1^L + \frac{3}{10} (c_1 + 1) \mathbf{b}_2^L + \frac{c_1}{10} \mathbf{b}_3^L + \frac{1}{120} (s_1 + s_2) \mathbf{v}_0^L \\ &\quad + \frac{1}{60} (3s_1 + s_2) \mathbf{v}_1^L + \frac{s_1}{60} \mathbf{v}_2^L \\ \mathbf{s}_{(1,3,1)}^1 &= \mathbf{s}_{(2,3,0)}^1 - \frac{3}{20} \mathbf{b}_1^L + \frac{3}{40} (1 - 2c_1) \mathbf{b}_2^L + \frac{3}{40} (1 + 2c_1) \mathbf{b}_3^L + \frac{1}{40} (s_1 + s_2) \mathbf{v}_1^L + \frac{1}{80} (3s_1 + s_2) \mathbf{v}_2^L \\ \mathbf{s}_{(0,4,1)}^1 &= \mathbf{s}_{(1,4,0)}^1 - \frac{3}{10} \mathbf{b}_2^L + \frac{3}{10} \mathbf{b}_3^L + \frac{s_1}{20} (s_1 + s_2) \mathbf{v}_2^L \end{aligned} \quad (\text{A.12})$$

et

$$\begin{aligned} \mathbf{s}_{(4,0,1)}^2 &= \mathbf{s}_{(5,0,0)}^2 - \frac{3}{10} \mathbf{b}_0^R + \frac{3}{10} \mathbf{b}_1^R + \frac{s_1}{20} (s_1 + s_2) \mathbf{v}_0^R \\ \mathbf{s}_{(3,1,1)}^2 &= \mathbf{s}_{(4,1,0)}^2 - \frac{3}{40} (3 - 2c_2) \mathbf{b}_0^R + \frac{3}{40} (1 - 2c_2) \mathbf{b}_1^R + \frac{3}{20} \mathbf{b}_2^R + \frac{1}{80} (s_1 + 3s_2) \mathbf{v}_0^R + \frac{1}{40} (s_1 + s_2) \mathbf{v}_1^R \\ \mathbf{s}_{(2,2,1)}^2 &= \mathbf{s}_{(3,2,0)}^2 - \frac{1}{20} \mathbf{b}_0^R - \frac{1}{10} (2 - c_2) \mathbf{b}_1^R + \frac{1}{20} (5 - 4c_2) \mathbf{b}_2^R + \frac{1}{20} \mathbf{b}_3^R + \frac{s_2}{60} \mathbf{v}_0^R \\ &\quad + \frac{1}{60} (s_1 + 3s_2) \mathbf{v}_1^R + \frac{1}{120} (s_1 + s_2) \mathbf{v}_2^R \\ \mathbf{s}_{(1,3,1)}^2 &= \mathbf{s}_{(2,3,0)}^2 - \frac{3}{10} (1 - c_2) \mathbf{b}_1^R + \frac{3}{40} (1 - 2c_2) \mathbf{b}_2^R + \frac{3}{40} (3 - 2c_2) \mathbf{b}_3^R + \frac{s_2}{20} \mathbf{v}_1^R + \frac{1}{80} (s_1 + 3s_2) \mathbf{v}_2^R \\ \mathbf{s}_{(0,4,1)}^2 &= \mathbf{s}_{(1,4,0)}^2 - \frac{3}{5} (1 - c_2) \mathbf{b}_2^R + \frac{3}{5} (1 - c_2) \mathbf{b}_3^R + \frac{s_2}{10} \mathbf{v}_2^R. \end{aligned} \quad (\text{A.13})$$

Les points de contrôle correspondant aux frontières E_2 et E_3 sont obtenus par décalage cyclique des indices dans A.7, A.12 et A.13 d'une unité pour la frontière $E_2(u)$ et de deux unités pour la frontière $E_3(u)$.

Annexe B

Structures des données dans le programme interactif

B.1 Structures de données

Les trois structures utilisées pour stocker les données du réseau surfacique sont les suivantes :

- Le type utilisé pour stocker les sommets est

```
typedef struct s_Vertex{
    double V[SPACEDIM]; /* les coordonnées du sommet */
    double N[SPACEDIM]; /* les coordonnées du vecteur normal */
    double XU[SPACEDIM], XV[SPACEDIM]; /* pointeur vers les */
                                        /* vecteurs de la base */
                                        /* du plan tangent */
    double *R2BI; /* pointeur vers l'antécédent du */
                /* vecteur des dérivées secondes */
    int F; /* indice d'une des faces incidentes */
        /* au sommet */
    int O; /* degré (valence, ordre) du sommet */
    double alpha, smooth; /* le paramètre de l'interpolation */
                        /* (alpha = 1 => interpolation) */
} Vertex;
```

- Le type utilisé pour stocker les arêtes est

```
typedef struct s_Edge{
    int V[2];
    char Border;
    double TARGET[SPACEDIM], TARGETDER[SPACEDIM];
```



```
double VMiddle[SPACEDIM];
    } Edge;
```

– le type utilisé pour stocker les faces est

```
typedef struct s_Face{
    int V[3];          /* les indices des trois sommets */
    int E[3];          /* les indices des trois aretes */
    int F[3];          /* les indices des trois faces voisines */
    double N[3];       /* la normale à la face */
    BP B[66];          /* les 66 points de contrôle des
                        4 sous-facettes quintiques */
    double *SP[4];     /* 4 pointeurs vers les facettes et
                        leurs normales (utilisés pour la
                        visualisation) */
} Face;
```

B.2 Format de stockage des réseaux surfaciques: le format NeiGhBour

Le programme d'implémentation des méthodes des chapitres 3, 4 et 5 commence par lire le réseau surfacique en entrée. Le réseau est sauvegardé dans un fichier de format spécial, appelé format NeiGhbour (ou simplement format NGB), qui signifie que chaque face pointe sur ses faces voisines. En effet, lors de la lecture, les sommets, les arêtes et les faces sont référencés dans 3 tableaux par l'intermédiaire des trois types de structure donnés ci-dessus. Une fois remplis, ces tableaux ne changeront pas ensuite. L'accès à un objet est possible grâce à son index dans le tableau correspondant. Un fichier au format NGB contient, dans son entête, deux nombres désignant respectivement le nombre des sommets et le nombre des faces. Juste après ces deux nombres, deux listes se suivent. La première contient les sommets du réseau. Chaque sommet est écrit dans une ligne séparée, sous le format :

```
<x> <y> <z>
```

où <x>, <y> et <z> sont les coordonnées x, y et z du sommet. La numérotation des sommets commence implicitement à 0.

La seconde liste contient l'information topologique : l'ensemble des faces et les adjacences entre faces. Le format d'une face est

```
<sid1> <sid2> <sid3> <fid1> <fid2> <fid3>.
```

<sid1>, <sid2> et <sid3> sont les indices des trois sommets composant la face. L'ordre de ces sommets est important pour l'orientation de la face. <fid1>, <fid2> et <fid3> sont les indices des trois faces opposées aux sommets <sid1>, <sid2> et <sid3>.

Bibliographie

- [1] P. Alfeld. *A trivariate Clough-Tocher scheme for tetrahedral data*. MRC Technical Summary Report, Mathematics Research Center, University of Wisconsin-Madison, 1984.
- [2] P. Alfeld. *Scattered Data Interpolation in Three or More Variables*. in *Mathematical Methods in Computer Aided Geometric Design*, Tom Lyche, Larry L. Schumaker (eds.), 1–34, Academic Press, New York, 1989.
- [3] L. Piegl and. On NURBS: a survey. *Computer Graphics and Applications*, 11(1):55–71, 1991.
- [4] C. L. Bajaj and I. Ihm. Smoothing polyhedra using implicit algebraic splines. *Computer graphics*, 26:79–88, 1992.
- [5] R. Barnhill, G. Birkhoff, and W. Gordon. Smooth interpolation in triangles. *Journal of Approximation Theory*, 8(2):114–128, 1973.
- [6] P. Bézier. Définition numérique des courbes et surfaces i. *Automatisme*, 11:625–632, 1966.
- [7] P. Bézier. Définition numérique des courbes et surfaces ii. *Automatisme*, 12:17–21, 1967.
- [8] P. Bézier. *Mathematical and practical possibilities of UNISURF*. in R.E. Barnhill, R. Riesenfeld (ed.): *Computer Aided Geometric Design*, pages 127–152. Academic Press, 1974.
- [9] P. Bézier. *Essai de définition numérique des courbes et des surfaces expérimentales*. PhD thesis, Université de Paris VI, 1977.
- [10] P. Bézier. *The mathematical Basis of the UNISURF CAD System*. Butterworth, 1986.
- [11] H. Biermann, Adi Levin, and Denis Zorin. Piecewise smooth subdivision surfaces with normal control. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 113–120. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [12] M.I.G. Bloor and M.J. Wilson. Spectral approximations to pde surfaces. *Computer-Aided Design*, 28(2):145–152, 1996.

- [13] W. Boehm. *Generating the Bézier points of triangular splines, in Surfaces in Computer Aided Geometric Design : Barnhill R. F., Boehm W. (ed.), pages 77–91.* North Holland Publishing Company, 1983.
- [14] W. Boehm. Geometric concepts for geometric design. *SIAM Review*, 37(3):473–??, 1995.
- [15] W. Boehm, G. Farin, and J. Kahmann. A survey of curve and surface methods in CAGD. *Comput. Aided Geometric Design*, 1:1–60, 1986.
- [16] H. Bohl and U. Reif. Degenerate Bézier patches with continuous curvature. *Computer Aided Geometric Design*, 14:749–761, 1997.
- [17] G.-P. Bonneau. *Variational Design of Rational Bézier Curves and Surfaces.* PhD thesis, Fachbereich Informatik der Universität Kaiserslautern, 1993.
- [18] G. P. Bonneau and H. Hagen. *Variational design of rational Bézier curves and surfaces, in Curves and Surfaces in Geometric Design, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), pages 51-58.* A. K. Peters, Wellesley MA, 1994.
- [19] G.-P. Bonneau and H. Hagen. *Variational design of rational Bézier curves and surfaces, in Curves and Surfaces II, pages 51-58.* Laurent, Le Méhauté, Schumaker (eds.), 1994.
- [20] G. Brunnet, H. Hagen, and P. Santarelli. Variational design of curves and surfaces. *Surv. Math. Ind.*, 3:1–27, 1993.
- [21] H. G. Burchard, J. A. Ayers, W. H. Frey, and N. S. Sapidis. *Approximation with Aesthetic Constraints, in Designing Fair Curves and Surfaces, pages 3–28.* S. Sapidis.(eds.), 1990.
- [22] H. Chiyokura and F. Kimura. Design of solids with Free-form surfaces. *Computer Graphics*, 17(3):289–298, 1983.
- [23] E. Coehn, T. Lyche, and L. L. Schumaker. Algorithms for Degree-Raising of splines. *ACM Transactions on Graphics*, pages 171–181, July 1985.
- [24] S. Coons. *Surfaces for computer aided design of space forms. Technical report.* MIT. Projet MAC-TR 41., 1967.
- [25] S. Dahlke, W. Dahmen, I. Weinreich, and E. Schmitt. Multiresolution analysis and wavelets on s^2 and s , 1995.
- [26] P. Davis. *Circulant Matrices.* Wiley, 1979.
- [27] C. DeBoor, K. Höllig, and M. Sabin. High accuracy geometric hermite interpolation. *Computer Aided Geometric Design*, 4:269–278, 1987.
- [28] W. L. F. Degen. Explicit continuity conditions for adjacent Bézier surface patches. *Computer Aided Geometric Design*, 7:181–189, 1987.
- [29] T. D. DeRose, M. Lounsbery, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. Technical Report TR-93-10-05, 1993.

- [30] W. Du and F. Schmitt. *A general method of treating degenerate Bézier patches*, in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté and L.L. Schumaker (eds.), pages 161–164. Academic Press, Boston, 1991.
- [31] W. H. Du. *Etude sur la représentation de surfaces complexes: application à la reconstruction de surface échantillonnées*. PhD thesis, Department Image, TELECOM, Paris, 1988.
- [32] N. Dyn, J.A. Gregory, and D. Levin. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graphics*, 9:160–169, April 1990.
- [33] N. Dyn, S. Hed, and D. Levin. Subdivision schemes for surface interpolation. *Workshop in Computational Geometry*, pages 97–118, 1993.
- [34] N. Dyn, D. Levin, and J.A. Gregory. A four-point interpolatory subdivision scheme interpolation for curve design. *Computer-Aided Geometric Design*, 4(4):257–268, December 1987.
- [35] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multi-resolution analysis of arbitrary meshes. *Computer Graphics*, 29(Annual Conference Series):173–182, 1995.
- [36] M. Eck and H. Hoppe. Automatic reconstruction of B-spline surfaces of arbitrary topological type. *Computer Graphics (SIGGRAPH)*, pages 325–334, 1996.
- [37] G. Farin. A construction for visual C^1 continuity of polynomial surface patches. *Computer Graphics and Image Processing*, 20:272–282, 1982.
- [38] G. Farin. *Some aspects of car body design at Daimler-Benz*. in *Surfaces in Computer Aided Geometric Design*, R. Barnhill, W. Boehm (eds.), North-Holland, 1982.
- [39] G. Farin. *Smooth interpolation to scattered 3D data*, in *Barnhill R. F., Boehm W. (ed.)*, pages 43–63. North Holland Publishing Company, 1983.
- [40] G. Farin. A modified Clough-Tocher interpolant. *Computer Aided Geometric Design*, 2:19–27, 1985.
- [41] G. Farin. Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design*, 3:83–127, 1986.
- [42] G. Farin. *Rational Curves and Surfaces*. in *Mathematical Methods in Computer Aided Geometric Design*, Tom Lyche, Larry L. Schumaker (eds.), pages 215–238, Academic Press, New York, 1989.
- [43] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design*. Academic Press, New York, 4th edition, 1997.
- [44] R. Farouki and V. Rajan. On the numerical condition of polynomials in bernstein form. *Computer Aided Geometric Design*, 4(3):191–216, 1987.
- [45] R. T. Farouki and T. N. T. Goodman. On the optimal stability of the Bernstein basis. *Mathematics of Computation*, 65(216):1553–1566, 1996.

- [46] G. Fasshauer and L. Schumaker. *Scattered data fitting on the sphere*. in Mathematical Methods for Curves and Surfaces II, M. Daehlen, T. Lyche, and L. L. Schumaker (eds.), pages 117–166, Vanderbilt Univ. Press, Nashville, 1998.
- [47] I. Faux and M. Pratt. *Computational Geometry for Design and Manufacture*. Ellis Horwood, 1979.
- [48] D. Fergusson. Construction of curves and surfaces using numerical optimisation techniques. *Computer-Aided Design*, 18:15–21, 1986.
- [49] J. D. Foley, A. van Dam, Steven K. Feiner, and John F. Hughes. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Publishing Company, 1990.
- [50] P. Fong and HP. Seidel. An implementation of triangular B-spline surfaces over arbitrary triangulations. *Computer Aided Geometric Design*, 10:267–275, 1993.
- [51] R. Franke. *Scattered data interpolation: Tests of some methods*. *Mathematics of Computation*, 38:181–200, 1982.
- [52] R. Franke and G. Nielson. *Scattered data interpolation and Applications: A tutorial and Survey*, in *Geometric Modeling, Survey on SD methods*,. H. Hagen, D. Roller (eds.), pages 131-159, 1990.
- [53] W. Freeden and M. Schreiner. Orthogonal and Non-orthogonal multiresolution analysis: Scale discrete and exact fully discrete expansions on the sphere. *AGTM-Report., University of Kaiserslautern, Geomathematics Group*, Nr. 1637, 1996.
- [54] W. Freeden, M. Schreiner, and R. Franke. A survey on spherical spline approximation. *Surveys Math. Indust.*, 7:29–85, 1997.
- [55] R. N. Goldman. Subdivision algorithms for Bézier triangles. *Computer Aided Design*, 15(3):159–166, 1983.
- [56] J. A. Gregory. *Smooth interpolation without twist constraints*. in R. E. Barnhill, R.E. Riesenfeld (ed.): *Computer Aided Geometric Design*, pages 71–88. Academic Press, 1974.
- [57] J. A. Gregory. C^1 rectangular and non-rectangular surface patches. in R. E. Barnhill, W. Boehm (ed.): *Surfaces in Computer Aided Geometric Design*, pages 25–33. North-Holland, 1983.
- [58] J. A. Gregory. *N-sided surface patches*, in *The Mathematics of Surfaces J. Gregory (ed.)*, pages 217–232. Clarendon Press, Oxford, 1986.
- [59] J.A Gregory and P. Charrot. A C^1 triangular interpolation patch for computer-aided geometric design. *Computer Graphics and Image Processing*, 13:80–87, 1980.
- [60] J.A. Gregory and J. Hahn. Geometric continuity and convex combination patches. *Computer Aided Geometric Design*, 4:79–89, 1987.
- [61] E. Guilbert. *Interpolation G^1 Hiérarchique*. Mémoire de D.E.A, Univ. Joseph Fourier, Grenoble, 2000.

- [62] H. Hagen. Geometric surface patches without twist constraints. *Comput. Aided Geometric Design*, 3:179–184, 1986.
- [63] H. Hagen and H. Pottmann. *Curvature continuous triangular interpolants*. in *Mathematical Methods in Computer Aided Geometric Design*, Tom Lyche, Larry L. Schumaker (eds.), pages 373–384, Academic Press, New York, 1989.
- [64] H. Hagen and G. Schulze. Automatic smoothing with geometric surface patches. *Computer Aided Geometric Design*, 4(3):231–235, 1987.
- [65] S. Hahmann and G.-P. Bonneau. Triangular G^1 interpolation by 4-splitting domain triangles. *Computer Aided Geometric Design*, 17:731–757, 2000.
- [66] S. Hahmann, G.-P. Bonneau, and R. Taleb. Smooth irregular mesh interpolation, in curve and surface fitting: Saint-malo 1999, Albert Cohen, Christophe Rabut, and Larry L. Schumaker (eds.), pages 237–246. pages 237–246, 2000.
- [67] S. Hahmann, G.-P. Bonneau, and R. Taleb. Localizing the 4-split method for G^1 free-form surface fitting. *to appear in Computing Suppl.*, 2001.
- [68] S. Hahmann, G.-P. Bonneau, and R. Taleb. Modélisation de surfaces de topologie arbitraire. *Actes des Journées Modélisation Géométriques, Dijon*, Mars 2001.
- [69] D. Hansford. *Boundary Curves with Quadric Precision for Tangent Continuous Scattered Data Interpolant*. PhD thesis, Arizona State University de Paris VI, 1991.
- [70] R. L. Hardy. *Multiquadric equations of topography and other irregular surfaces*. *Journal Geophys. Research*, 76:1905–1915, 1971.
- [71] T. Hermann. G^2 interpolation of free form curve networks by biquintic gregory patches. *Computer Aided Geometric Design*, 13:873–893, 1996.
- [72] G. Herron. Smooth closed surfaces with discrete triangular interpolants. *Computer Aided Geometric Design*, 2:297–306, 1985.
- [73] C. Hoffmann. *Geometric and solid modeling*. Morgan Kaufmann, 1989.
- [74] C. Hoffmann. Implicit curves and surfaces in cagd. *IEEE Computer Graphics and Applications*, 13(1):79–88, 1993.
- [75] K. Höllig and J. Koch. Geometric hermite interpolation. *Computer Aided Geometric Design*, 12:567–580, 1995.
- [76] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. A.K. Peters, 1993.
- [77] Andreas Hubeli and Markus Gross. A survey of surface representations for geometric modeling. techreport 335, ETH Zurich s ETH Zurich, Institute of Scientific Computing, March 2000.
- [78] T. Jensen. *Assembling triangular and rectangular patches and multivariate splines*. in *Geometric Modeling: Algorithms and new Trends*, G. Farin (ed.), pages 203–220, SIAM, 1987.

- [79] A. K. Jones. Nonrectangular surface patches with curvature continuity. *Computer-Aided Design*, 20:325–335, 1988.
- [80] C. Lawson. surface interpolation for scattered data on a sphere. *Rocky Mountain J. Math.*, 14:177–202, 1984.
- [81] Jean-Claude Léon. *Modélisation et construction de surfaces pour la CFAO*. Paris: Hermes, 1991.
- [82] D. Liu and J. Hoschek. GC^1 continuity conditions between adjacent rectangular and triangular Bézier surface patches. *Computer-Aided Design*, 21:194–200, 1989.
- [83] M. Lizhuang and Q. Peng. Smoothing of free-form surfaces with Bézier patches. *Computer Aided Geometric Design*, 12:231–249, 1995.
- [84] C. Loop. A G^1 triangular spline surface of arbitrary topological type. *Computer Aided Geometric Design*, 11:303–330, 1994.
- [85] C. Loop and T. DeRose. Generalized B-spline surfaces of arbitrary topology. *Computer Graphics*, 24:347–356, 1990.
- [86] J. M. Lounsbery. *Multiresolution Analysis for Surfaces of Arbitrary Topological type*. PhD thesis, Department of Computer Science and Engineering, University of Washington, 1994.
- [87] Michael Lounsbery, Tony D. DeRose, and Joe Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, 1997.
- [88] T. Lyche and L. Schumaker. A multiresolution tensor spline method for fitting functions on the sphere. *SIAM J. Sci. Comp.*, 22:724–747, 2000.
- [89] S. Mann. *Surface approximation using geometric Hermite patches*. PhD thesis, University of Washington, 1992.
- [90] S. Mann. *Using Local optimization in surface fitting*, in *Mathematical Methods for Curves and Surfaces*, Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds.), <http://citeseer.nj.nec.com/mann95using.html>. Vanderbilt University Press, Nashville & London, 1995.
- [91] S. Mann. Cubic precision Clough-Tocher interpolation. *Computer Aided Geometric Design*, 16:85–88, 1999.
- [92] S. C. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, and K. Sloan. *A survey of parametric scattered data fitting using triangular interpolants*, in *Curve and Surface Design*, H. Hagen (ed.), pages 145–172. SIAM, 1992.
- [93] D. S. Meek and D. J. Walton. On surface normal and gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design*, 17:521–543, 2000.

- [94] R. Mencl and H. Muller. *Interpolation and approximation of surfaces from three-dimensional scattered data points*, in *Scientific Visualization, Dagstuhl'97*. IEEE Publishers, 2000.
- [95] H. P. Moreton. *Minimum Curvature Variation Curves, Networks, and Surfaces for Fair Free-Form Shape Design*. PhD thesis, University of California at Berkeley, 1993.
- [96] H. P. Moreton and C. H. Séquin. *Surface Design with Minimum Energy Networks*, in *Proceedings Symposium on Solid Modeling Foundations and CAD/CAM Applications*, ed. J. Rossignac and J. Turner, pages 291–301. ACM Press, New York, 1991.
- [97] H. P. Moreton and C. H. Séquin. Functional optimisation for fair surface design. *Computer Graphics*, 26(2):167–176, July 1992.
- [98] M. Neamtu and F. Pfluger. Degenerate polynomial patches of degree 4 and 5 used for geometrically smooth interpolation in \mathbb{R}^3 . *Computer Aided Geometric Design*, 11:451–474, 1994.
- [99] G. Nielson. *Some Piecewise Polynomial Alternatives to Splines Under Tension*, in *Computer Aided Geometric Design*, ed. by Barnhill/Reienfeld, pages 209–236. Academic Press, 1974.
- [100] G. Nielson. Minimum norm interpolation in triangles. *SIAM J. Numer. Anal.*, 17(1):44–62, 1980.
- [101] G. Nielson and R. Ramaraj. Interpolation over a sphere. *Computer Aided Geometric Design*, 4:41–57, 1987.
- [102] G. M. Nielson. The side-vertex method for interpolating in triangles. *Journal of Approximation theory*, 25:318–336, 1979.
- [103] G. M. Nielson. A method for interpolating scattered data based upon a minimum norm network. *Mathematics of Computation*, 40:253–271, 1983.
- [104] G. M. Nielson. *A Transfinite, Visually Continuous Triangular Interpolant*. in G. Farin (ed.): *Geometric Modeling, Algorithms and New Trends*, pages 235–246. SIAM, 1987.
- [105] H. Nowacki and D. Reese. *Design and Fairing of Ship Surfaces*, in *Surfaces in Computer Aided Geometric Design: Barnhill R. F., Boehm W. (ed.)*, pages 121–134. North Holland Publishing Company, 1983.
- [106] J. Peters. Local cubic and bicubic C^1 surface interpolation with linearly varying boundary normal. *Comput. Aided Geometric Design*, 7:499–516, 1990.
- [107] J. Peters. Local smooth surface interpolation: a classification. *Computer Aided Geometric Design*, 7:191–195, 1990.
- [108] J. Peters. Smooth interpolation with cubic patches. *Comput. Aided Design*, 22(2):109–120, 1990.

- [109] J. Peters. Parametrizing singularly to enclose vertices by a smooth parametric surface. In Kurt Akeley, editor, *Proceedings of Graphics Interface '91*, S. MacKay, E.M. Kidd (eds.), pages 1–7. Canadian ManComputer Communications Society, 1991.
- [110] J. Peters. Smooth interpolation of a mesh of curves. *Constructive Approximation*, 7:221–246, 1991.
- [111] L. Piegl and W. Tiller. *The NURBS book*. Springer-Verlag, 1995.
- [112] D. Pilcher. *Smooth Parametric Surfaces*, in *Computer Aided Geometric Design*, ed. by Barnhill/Reienfeld, pages 237–253. Academic Press, 1974.
- [113] B. R. Piper. *Visually smooth interpolation with triangular Bézier patches*. in *Geometric Modeling: Algorithms and new Trends*, G. Farin (ed.), SIAM, pages 221–233, 1987.
- [114] Robert J. Renka. Interpolation of data on the surface of a sphere. *ACM Transactions on Mathematical Software*, 10(4):417–436, 1984.
- [115] J. Risler. *Mathematical Models for CAD*. Masson, Paris, 1992.
- [116] J. Roulier and T. Rando. *Measures of Fairness for Curves and Surfaces*, in, S. Sapidis.(eds.), pages 75–122, 1990.
- [117] M. A. Sabin. *The use of potentiel surfaces for numerical geometry*. B.A.C. Weybridge, VTO/MS/153, 1968.
- [118] R. F. Sarraga. G^1 interpolation of generally unrestricted cubic Bézier curves. *Computer Aided Geometric Design*, 4:23–39, 1987.
- [119] Peter Schröder and Wim Sweldens. Spherical Wavelets: Efficiently Representing Functions on the Sphere. In *Computer Graphics Proceedings, Annual Conference Series, 1995 (ACM SIGGRAPH '95 Proceedings)*, pages 161–172, 1995.
- [120] L. L. Schumaker. *Reconstructing 3D objects from cross sections*, in Dahmen, W. Gasca M. and Micchelli, C.A., eds., *Computation of Curves and Surfaces*, pages 275–309. Cluwer, Dordrecht, 1990.
- [121] D. Schweikert. An interpolation curve using a spline in tension. *Journal of Math. and Physics*, 45:312–317, 1988.
- [122] T. Sederberg. Piecewise algebraic surface patches. *Comput. Aided Geometric Design*, 2:53–59, 1985.
- [123] H-P. Seidel. A new multiaffine approach to B-splines. *Comput. Aided Geometric Design*, 6:23–32, 1989.
- [124] L.A. Shirman and C.H. Séquin. Local surface interpolation with Bézier patches. *Computer Aided Geometric Design*, 4:279–295, 1987.
- [125] L.A. Shirman and C.H. Séquin. Local surface interpolation with Bézier patches: errata and improvements. *Computer Aided Geometric Design*, 8:217–221, 1991.

- [126] K. Sloan. Surface normal (summary). *Usenet comp. graphics article*, june 1992.
- [127] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998.
- [128] P. H. Todd and R. J. Y. McLeod. Numerical estimation of the curvature of surfaces. *Computer-Aided Design*, 18(1):33–37, 1986.
- [129] J.J. Van Wijk. Bicubic patches for approximating non-rectangular control meshes. *Computer Aided Geometric Design*, 3:1–13, 1986.
- [130] A. WATT. *3D Computer Graphics*. Addison-Wesley, 1993.
- [131] M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide, second edition*. Addison-Wesley, 1998.
- [132] Y. Xiuzi. Generating Bézier points for curves and surfaces from boundary information. *Computer-Aided Design*, vol. 27 No. 12:875–885, 1995.
- [133] Y. Xiuzi. The gaussian and mean curvature criteria for curvature continuity between surfaces. *Computer Aided Geometric Design*, 13:549–567, 1996.
- [134] Y. Xiuzi. Curvature continuous interpolation of curve meshes. *Computer Aided Geometric Design*, 14:169–190, 1997.
- [135] Y. Xiuzi, Y. Liang, and H. Nowacki. Geometric continuity between adjacent Bézier patches and thier constructions. *Computer Aided Geometric Design*, 13:521–548, 1996.
- [136] D. Zorin. *Stationary Subdivision and Multiresolution Surface Representations*. PhD thesis, California Institut of Technology, Pasadena, 1998.
- [137] D. Zorin and W. Schröder, P. Sweldens. Interactive multiresolution mesh editing. *Computer Graphics*, 31(SIGGRAPH '97 Proceedings):259–268, 1997.
- [138] Denis Zorin, Peter Schröder, and Wim Sweldens. Interpolating subdivision for meshes with arbitrary topology. *Computer Graphics*, 30(Annual Conference Series):189–192, 1996.